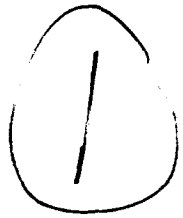


Carnegie-Mellon University
Software Engineering Institute



Kernel User's Manual
Version 3.0

Appendix A: Ada Code

Judy Bamberger
Tim Coddington
Robert Firth
Daniel Klein
David Stinchcomb
Roger Van Scoy
December 1989

AD-A228 514

DTIC
ELECTE
NOV 05 1990
S D
FD

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

User's Guide

CMU/SEI-89-UG-1

December 1989

Kernel User's Manual Version 3.0

Appendix A: Ada Code



Accession	
NTIS	✓
DTIC	TAB
Unannounced	
Justification	
By _____	
Distribution	
Availability	
Dist	Availability
A-1	Source

Judy Bamberger
Tim Coddington
Robert Firth
Daniel Klein
David Stinchcomb
Roger Van Scoy

Distributed Ada Real-Time Kernel Project

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

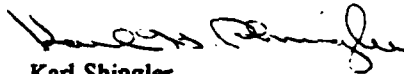
SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



Karl Shingler
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1989 Carnegie Mellon University

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Service. For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Table of Contents

package Hardware_Interface	6	package St	162
generic package Generic_Kernel_Time	18	package Tg	162
package Kernel_Time	45	package Communication_Management	192
generic package Generic_Time_Globals	49	generic package Generic_Process_Attribute_Modifiers	195
package Time_Globals	83	package Ke	195
generic package Generic_Schedule_Types	86	package Plb	195
package Schedule_Types	91	package St	195
generic package Generic_Network_Globals	94	package Tg	196
package Network_Globals	99	package Process_Attribute_Modifiers	206
generic package Generic_Network_Configuration	102	generic package Generic_Process_Attribute_Readers	210
package Network_Configuration	110	package Plb	210
generic package Generic_Processor_Management	118	package Ke	210
package Ke	118	package St	210
package Tg	118	package Process_Attribute_Readers	217
package Processor_Management	128	generic package Generic_Interrupt_Globals	220
generic package Generic_Process_Managers_Globals	131	package Interrupt_Globals	227
package Process_Managers_Globals	134	generic package Generic_Interrupt_Management	230
generic package Generic_Process_Managers	140	package Ig	230
package Process_Managers	152	package Interrupt_Management	241
generic package Generic_Communication_Globals	155	generic package Generic_Semaphore_Management	244
package Communication_Globals	157	package Semaphore_Management	255
generic package Generic_Communication_Management	162	generic package Generic_Alarm_Management	258
package Ke	162	package Ke	258
package Cg	162	package St	258
package Ng	162	package Tg	258
package Plb	162	package Alarm_Management	266
		generic package Generic_Time_Management	269

package Time_Management	281
generic package Generic_Timeslice_Management	284
package Tg	284
package Ke	285
package Timeslice_Management	289
package Context_Save_Area	293
generic package Generic_Process_Table	304
package Time_Event_Queue	308
package Process_Table	333
package Process_Table	341
generic package Generic_Tool_Interface	345
package Cg	345
package Ig	345
package Ke	345
package Ng	345
package Plb	345
package St	345
package Tg	345
package Tlg	345
package Tool_Interface	357
package Tool_Interface_Output	359
package Hardware_Interface	364
generic package Generic_Kernel_Time	378
package Kernel_Time	404
Index	427

[illegible][illegible][illegible]

```

../ .....
../ MODULE NAME
../ hardware_interface
../
../ MODULE TYPE
../ package_specification
../
../ MODULE PURPOSE:
../ The Kernel package hardware_interface
../ provides an interface to compiler-specific primitive types.
../ Within the Kernel itself, there are no references to the
../ predefined types in Ada package Standard;
../ all references to primitive types use names declared in
../ package hardware_interface
../ By doing this, certain implementation-dependent details are abstracted
../ away from both the Kernel and the application in a uniform manner.
../ It is recommended that applications avoid using package Standard entirely
../ and use package hardware_interface for ready compatibility with
../ Kernel primitives.
../
../ .....
../ MODULE DESCRIPTION:
../ these types are defined in case the compiler's definition and/or
../ implementation of them is not sufficient for the Kernel and/or the
../ application; this allows Kernel/application developers to provide
../ their own version of types and operators without needing to modify
../ Kernel or application code directly
../
../ MODULE CONTENTS:
../ hardware_interface (package specification)
../ bits_per_byte (constant)
../ byte (constant)
../ hw_address (type)
../ hw_bis8 (type)
../ hw_bis8_ptr (type)
../ hw_byte (type)
../ hw_byte_ptr (type)
../ hw_duration (type)
../ hw_integer (type)
../ hw_long_integer (type)
../ hw_long_natural (type)
../ hw_long_positive (type)
../ hw_natural (type)
../ hw_positive (type)
../ hw_short_integer (type) (CURRENTLY COMMENTED OUT)
../ hw_string (type)
../ longword (constant)
../ null_hw_address (constant)
../ to_hw_address (function)
../ to_hw_bis8_ptr (function)
../ to_hw_byte_ptr (function)
../ word (constant)
../
../ REFERENCES:

```

```

../ DESIGN DOCUMENTS:
../ Kernel Facilities Definition
../ --REQ: 5.1.4
../ --PRIM: N/A
../ --TIME: N/A
../
../ USER'S MANUAL:
../ Kernel User's Manual
../
../ TESTING AND VALIDATION:
../ none
../
../ NOTES:
../ none
../
../ .....
../ MODIFICATION HISTORY:
../ 18aug88 bamberg created (complies clean)
../ 30aug88 bamberg correct typo on bound of hw_integer
../ 31aug88 dvk changed 'byte' to 'bits_per_byte'
../ 14sep88 dvk added constants byte, word, and longword
../ 17oct88 firh added hw_long_natural
../ hw_bis8_ptr and conversions
../ 1Nov88 firh added to hw_address and null_hw_address
../ 12Dec88 firh revised documentation; reorganised
../ 11Jan89 firh final doc. check for beta release
../ 25aug89 bamberg final doc. check for 01sep89 freeze
../
../ .....
../ VERSION: 3.0
../
../ DISCLAIMER:
../ The following copyright must be included in this software and
../ all software utilizing this software.
../
../ DISTRIBUTION:
../ Approved for public release; distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies, and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names

```

```

--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....

```

pragma (page);

```

with System;
-- to get: address

with Unchecked_Conversion;

--/ .....
--/ package short name: HI
--/ .....
package Hardware_Interface
is
--/ The following definitions might need modification after any change to
--/ either compiler or hardware. The purpose of the definitions is to
--/ provide a coordinated and clean view of the underlying basic machine
--/ data types and addressing structure.

--
-- define the bit interface to the underlying system

Bits_Per_Byte : constant := 8;

--
-- define constants to be used as offsets in rep specs; these values
-- indicate how many bytes in the named storage unit (i.e., byte, word,
-- longword)

--
-- a byte contains 1 byte
-- a word contains 2 bytes
-- a longword contains 4 bytes
--

Byte : constant := 1;
Word : constant := 2;
Longword : constant := 4;

-- The type hw_duration must capture exactly the hardware
-- representation of the current Ada type DURATION
--
-- to be defined for hw_duration:
-- range
-- size
-- 'small (i.e., granularity of measure)
--
-- In the event of any mismatch, the compiler should reject the
-- representation clauses

type Hw_Duration is new Duration range -86_400.0 .. +86_400.0;
for Hw_Duration'Small use 2.0 * (-14);
for Hw_Duration'Size use 4 * Bits_Per_Byte;
pragma (page);

```



```

-- The types hw_byte and hw_bis8 are to be used to define essentially
-- untyped data formats, such as those found in device interfaces.
--
-- hw_byte is intended to be an unsigned 8-bit byte
-- hw_bis8 is intended to be a byte considered as a sequence of bits
--
-- Note especially that the bit numbering in the type hw_bis8 should
-- be whatever is most convenient for subsequent use of the type. If
-- the bulk of the hardware and device documentation numbers the bits
-- a certain way, that way should be used in the type declaration.

```

```

type hw_byte is range 0..255;

```

```

-- NOTE: the Telesoft compiler rejects the following representation clause
-- it will allocate TWO bytes (16 bits) for simple hw_byte objects
-- However, it will correctly allocate ONE byte (8 bits) for record
-- components of type hw_byte. If asked to by a rep clause.

```

```

-- for hw_byte size use 1*bits_per_byte;

```

```

-- the same, considered as a sequence of bits

```

```

-- NOTE: WE FOLLOW THE MC68020 CONVENTION OF NUMBERING THE BITS
-- FROM THE RIGHT (bit0 = least significant)

```

```

-- THE COMPILER USES THE OPPOSITE CONVENTION, SO THE REP SPEC
-- MUST REVERSE THE NUMBERING

```

```

type hw_bis8 is record

```

```

  bit7 : Boolean;
  bit6 : Boolean;
  bit5 : Boolean;
  bit4 : Boolean;
  bit3 : Boolean;
  bit2 : Boolean;
  bit1 : Boolean;
  bit0 : Boolean;
end record;

```

```

for hw_bis8 use record

```

```

  bit7 at 0 range 0..0;
  bit6 at 0 range 1..1;
  bit5 at 0 range 2..2;
  bit4 at 0 range 3..3;
  bit3 at 0 range 4..4;
  bit2 at 0 range 5..5;
  bit1 at 0 range 6..6;
  bit0 at 0 range 7..7;
end record;

```

```

pragma (page);

```

```

--/ =====
--/ SUBPROGRAM NAME:
--/   to_hw_bis8
--/
--/ DESCRIPTION:
--/   conversion routine from hw_byte to hw_bis8
--/
--/ --REQ: 5.1.4
--/ --PRIM: N/A
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/   Source
--/   an 8-bit value of type hw_byte
--/   (a small integer in the range 0..255)
--/
--/   default value:
--/   none
--/
--/   function return value
--/   the same bit pattern, considered as a record of eight
--/   single-bit components
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   return the same physical bit pattern as the parameter
--/   return the same physical bit pattern as the parameter
--/
--/ POSTCONDITIONS:
--/   NONE
--/
--/ ERROR PROCESSING:
--/   NONE
--/
--/ EXAMPLES:
--/   This conversion is to be used when it is necessary to inspect
--/   an individual bit of a device register or other hardware value.
--/
--/   Consider for instance a device status byte of which bit2 is the
--/   'ready' bit, asserted whenever the device is ready. The test
--/   in Ada then looks like this:
--/
--/   if to_hw_bis8(status_byte).bit2 then -- device is ready
--/
--/ =====

```

```

function To_Hw_Bis8 is new Unchecked_Conversion(hw_byte, hw_bis8);
pragma (page);

```



```

-- The integer types are intended to capture the natural efficient
-- integer types of the target machine. Currently, these are
--
-- 8-bit signed : hw_short_integer
-- 16-bit signed : hw_integer
-- 32-bit signed : hw_long_integer
--
-- These types are defined in terms of explicit ranges, and a size
-- clause is provided as a check that the compiler has indeed mapped
-- them onto the correct hardware type.
--
-- The Ada types defined in package STANDARD should not be used, since
-- they are dependent on the compiler as well as the target
--
-- hw_integer is intended to be a 16-bit integer
--
-- to be defined for hw_integer:
--
-- range
-- size
--
type hw_integer is range -32_768 .. 32_767;
for hw_integer'size use 2 * Bits_Per_Byte;

-- hw_short_integer is intended to be an 8-bit integer
--
-- to be defined for hw_short_integer:
--
-- range
-- size
--
type hw_short_integer is range -128 .. 127;
for hw_short_integer'size use 1 * Bits_Per_Byte;

-- hw_long_integer is intended to be a 32-bit integer
--
-- to be defined for hw_long_integer:
--
-- range
-- size
--
type hw_long_integer is range -2_147_483_648 .. 2_147_483_647;
for hw_long_integer'size use 4 * Bits_Per_Byte;
pragma (page);

```

```

-- SUBPROGRAM NAME:
-- to_hw_bits8_ptr
--
-- DESCRIPTION:
-- conversion routine from Ada address type to access type
--
-- REQ: 5.1.4
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- Source
-- the address of a declared object
-- WHICH MUST BE A HW_BYTE OR A HW_BITS8
--
-- default value:
-- none
--
-- function return value
-- the same bit pattern, considered as an access value designating
-- a hw_bits8
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- return the same physical bit pattern as the parameter
--
-- POSTCONDITIONS:
-- NONE
--
-- ERROR PROCESSING:
-- NONE
--
-- EXAMPLES:
-- see above
--
function To_Hw_Bits8_Ptr is new Unchecked_Conversion(hw_Byte_Ptr, Hw_Bits8_Ptr);
pragma (page);

```

```

.. For additional security, the appropriate subsets of the machine
.. integer types are defined as full types, not as subtypes. The
.. two subsets are
..
.. Natural: the non-negative half of the range (0..Last)
.. Positive: the strictly positive subset (1..Last)
..
.. Again, a size clause is added as a check.
..
.. hw_long_natural is intended to be a 32-bit integer with natural range
..
.. to be defined for hw_long_natural:
.. range
.. size
..
type hw_long_natural is range 0 .. 2 147 483 647;
for hw_long_natural'size use 4 * Bits_Per_Byte;

.. hw_long_positive is intended to be a 32-bit integer with positive range
..
.. to be defined for hw_long_positive:
.. range
.. size
..
type hw_long_positive is range 1 .. 2 147 483 647;
for hw_long_positive'size use 4 * Bits_Per_Byte;

.. the range of hw_natural is to be a subset of hw_integer
.. (i.e., only 16-bits)
..
.. to be defined for hw_natural:
.. range
.. size
..
type hw_natural is range 0 .. 32 767;
for hw_natural'size use 2 * Bits_Per_Byte;

.. the range of hw_positive is to be a subset of hw_integer
.. (i.e., only 16-bits)
..
.. to be defined for hw_positive:
.. range
.. size
..
type hw_positive is range 1 .. 32 767;
for hw_positive'size use 2 * Bits_Per_Byte;
pragma page;

```

```

.. hw_address is included just in the off chance that the compiler's
.. implementation of system_address is not acceptable to the Kernel and/or
.. the application; this allows Kernel/application developers to provide
.. their own version of address operators without needing to modify Kernel
.. or application code
..
.. There is also a function to convert an integer of the appropriate
.. size to an address, and a null value that cannot be a legal address
..
.. In the current implementation, it has been shown that the
.. compiler-provided address type is satisfactory.
..
type hw_address is new System_Address;

-- SUBPROGRAM NAME:
-- to_hw_address
--
-- DESCRIPTION:
-- conversion routine from an integer type to type hw_address
--
-- --REQ: NONE
--
-- --PRIM: NONE
--
-- --TIME: NONE
--
-- PARAMETERS:
-- Source
-- an integer value OF THE SAME SIZE AS AN ADDRESS
-- the value MUST be a legal address
--
-- default value:
-- none
--
-- function return value
-- the same bit pattern, considered as an address
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- return the same physical bit pattern as the parameter
--
-- POSTCONDITIONS:
-- NONE
--
-- ERROR PROCESSING:
-- NONE
--
-- EXAMPLES:
-- The most common use of this function is to construct a pointer
-- to a specific part of the machine's address space, for instance

```

```

--/ a part that corresponds to a device control register. Suppose
--/ that a device controlled by a set of registers whose addresses
--/ start at 16#1000ff00. A handler on this device can be created by
--/
--/ device c....: block address : hw address
--/ := to hw_address(16#1000ff00);
--/

```

```

function To_Hw_Address is
new unchecked_conversion(hw_long, Integer, hw_address);
Null_Hw_Address : constant hw_address
:= To_Hw_Address(hw_long, Integer'(0));

```

```

-- hw_string is included just in the off chance that the compiler's
-- implementation of string is not acceptable to the kernel and/or the
-- application, this allows kernel/application developers to provide their
-- own version of string operators without needing to modify kernel or
-- application code
--
-- In this initial implementation, we assume that the compiler-provided
-- string type and operations are OK

```

```

type hw_string is new string;
end hardware_interface;
pragma page;

```

```

--/ .....
--/ MODULE NAME
--/ generic_kernel_time
--/
--/ MODULE TYPE:
--/ generic_package_specification
--/
--/ MODULE PURPOSE:
--/ provide the abstraction of kernel time
--/
--/ .....
--/ MODULE DESCRIPTION:
--/ Kernel_time abstract type and subprograms to manipulate it
--/
--/ MODULE CONTENTS:
--/ generic_kernel_time (generic_package_specification)
--/ integral_duration (type)
--/ kernel_time (type)
--/ seconds (overloaded (2) function)
--/ milliseconds (function)
--/ microseconds (function)
--/ ticks_per_second (constant)
--/ ticks_per_second_value (generic formal parameter)
--/ zero_kernel_time (constant)
--/ "+" (unary, function)
--/ "*" (function)
--/ "-" (unary, function)
--/ ":" (function)
--/ "/" (function)
--/ "%" (function)
--/ "<" (function)
--/ "<=" (function)
--/ ">" (function)
--/ ">=" (function)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ Kernel_Facilities_Definition_Document
--/ REQ: 12.1.1, 12.1.5, 12.1.6, 12.1.25
--/ PRIM: N/A
--/ TIME: N/A
--/
--/ USER'S MANUAL:
--/ Kernel_User's_Manual
--/
--/ TESTING AND VALIDATION:
--/ Unit tests and integration testing with Time_keeper
--/
--/ NOTES:
--/ this package must be instantiated and used in place of the
--/ compiler provided Package_Calendar
--/
--/ This package provides an abstract data type representing Time
--/ for use by the kernel. It is at the lowest level of a set of
--/ abstractions that jointly respond to Requirements Section 12

```

```

--/
--/ .....
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (compiles clean)
--/ 18oct88 firh revised to interface to assembler
--/ 18oct88 firh changed type of ticks_per... to long int
--/ 24oct88 firh added 'milliseconds' and 'microseconds'
--/ 09jan89 firh improved documentation
--/ 11jan89 firh final doc. update for beta release
--/ 16feb89 firh corrected error in documentation
--/
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy or functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
with Hardware Interface; use Hardware Interface;
-- to get: hw duration
-- hw long integer
-- hw positive
pragma (page);

```

```

generic
--
-- the number of TICKS in a second (this value is never used within the
-- kernel except for the initialization of the constant, nor should it be
-- used without the Kernel)
--
Ticks_Per_Second_Value : In Hw_Long_Integer;
--/ .....
--/ package short name: GKT
--/ .....
package Generic_Kernel_Time
is
--
-- the constant number of TICKS per second (thus providing a mechanism of
-- converting from Kernel time to application time)
--
Ticks_Per_Second : Hw_Long_Integer := Ticks_Per_Second_Value;
--
-- type Kernel_time; the time on which elapsed time and epoch time
-- abstractions are built
--
-- this time representation allows applications beginning at time zero to
-- execute for some 150,000 years, so no overflow is monitored
--
-- .....
-- the user should ensure that adjusting any times does not approach the
-- limit of this time representation; proper Kernel functioning is NOT
-- guaranteed if time overflows
--
-- .....
Kernel_Time is represented as a signed 64-bit binary integer,
-- representing a count of microseconds. Hence, a kernel time
-- of 1,000,000 corresponds to one second.
--
-- For the purposes of Ada definition, kernel_time is a record of
-- two components, being respectively the low-order and high-order
-- 32 bits. The high-order component can be correctly declared to
-- be a signed 32-bit integer, of type hw_long_integer. The low-order
-- component, however, is properly an UNSIGNED 32-BIT INTEGER, which
-- this Ada compiler will not accept. Accordingly, it must be declared
-- to be SIGNED, which is wrong
--
-- The consequences are these
--
-- (a) If a kernel time value is printed in the "obvious" manner, ie by
-- printing each component, the low order part may be printed as a
-- negative number when in fact it is a large positive number

```

```

-- (b) If a kernel time value is constructed "by hand", ie as an
-- aggregate of two integers, the person writing the aggregate
-- must perform the necessary conversion from signed to unsigned
-- form. For example, a kernel time of "-1" is represented as
-- 16811911111111111111, ie as (-1,-1) in Ada terms.
--
-- Applications should use ONLY the exported constructor functions to
-- create values of type kernel_time or of types derived from it.

type Kernel_Time is record
  Low : Hw_Long_Integer := 0;
  High: Hw_Long_Integer := 0; -- NECESSARILY INCORRECT - see above
end record;

for Kernel_Time use record
  Low at 0 range 0..31;
  High at 4 range 0..31;
end record;

--
-- the value of zero for the Kernel_time abstraction
--
Zero_Kernel_Time : constant Kernel_Time := (0,0);

--
-- the range of durations represented as Integral values
--
type Integral_Duration is new Hw_Long_Integer;

```

```

pragma (page);

```

```

function Seconds1 (
  Ada_Duration : Hw_Duration)
return Kernel_Time;

function Seconds (
  Ada_Duration : Hw_Duration)
return Kernel_Time renames Seconds1;

pragma Interface(Assembly, Seconds1);
pragma Linkname(Seconds1, "GKT_seconds1");

--/ =====
--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Ada duration type to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Ada duration
--/ the time to convert in Ada duration format (note that this limits
--/ legal values to 0 to 86400 seconds, 1 day)
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input duration in Kernel_time format (note that this contains
--/ time up to 1 day)
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the Ada duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/
pragma (page);

```

```

function Seconds2 (
  An Integral Duration : Integral_Duration)
  return Kernel_Time;

function Seconds (
  An Integral Duration : Integral_Duration)
  return Kernel_Time renames Seconds2;

pragma Interface(Assembly, Seconds2);
pragma Linkname(Seconds2, "GKT_seconds2");

```

```

--/ =====
--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ an integral duration
--/ the time to convert in Integer format
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the integral duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/ =====

```

pragma (page);

```

function Milliseconds (
  Integral_Milliseconds : Integral_Duration)
  return Kernel_Time;

pragma Interface(Assembly, Milliseconds);
pragma Linkname(Milliseconds, "GKT_milliseconds");
--/ =====
--/ SUBPROGRAM NAME:
--/ milliseconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Integral_milliseconds
--/ the duration in milliseconds to convert to kernel_time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/ =====

```

pragma (page);


```

function Microseconds (
Integral_Microseconds : Integral_Duration)
return Kernel_Time;

pragma Interface(Assembly, Microseconds);
pragma Linkname(Microseconds, "GKT_microseconds");

--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Integral_microseconds
--/ the duration in microseconds to convert to kernel_time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the Input Integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the Integral duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```

pragma (page):

```

function Add (
Left : Kernel_Time;
Right : Kernel_Time)
return Kernel_Time;

function "+" (
Left : Kernel_Time;
Right : Kernel_Time)
return Kernel_Time renames Add;

pragma Interface(Assembly, Add);
pragma Linkname(Add, "GKT_add");

--/ SUBPROGRAM NAME:
--/ "+"
--/
--/ DESCRIPTION:
--/ adds two Kernel_time values returning a Kernel_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the addition operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the addition operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the addition operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical sum of the two Kernel_time values, returning a
--/ properly formatted Kernel_time result
--/
--/ POSTCONDITIONS:
--/ the sum of left and right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```

--/

pragma (page);

```
function Subtract (
  Left : Kernel Time;
  Right : Kernel Time)
return Kernel Time;

function "*" (
  Left : Kernel Time;
  Right : Kernel Time)
return Kernel Time renames Subtract;

pragma Interface(Assembly, Subtract);
pragma Linkname(Subtract, "GKT_subtract");
```

--/ SUBPROGRAM NAME:

--/

--/ DESCRIPTION:

--/ subtracts the right Kernel time value from the left Kernel time value

--/ value returning a Kernel time value

--/ --REQ: 12.1.23, 12.1.24

--/

--/ --PRIM: N/A

--/

--/ --TIME: N/A

--/

--/ PARAMETERS:

--/ left

--/ left operand of the subtraction operator (subtrahend)

--/

--/ default value:

--/ none

--/

--/ right

--/ right operand of the subtraction operator (minuend)

--/

--/ default value:

--/ none

--/

--/ function return value

--/ result of the subtraction operator

--/

--/ PRECONDITIONS:

--/ NONE

--/

--/ ACTIONS:

--/ takes the mathematical difference of the two Kernel time values by

--/ subtracting the right operand from the left operand, returning a

--/ properly formatted Kernel time result

--/

--/ POSTCONDITIONS:

--/ the difference of left and right is computed and returned

--/

--/ ERROR PROCESSING:

--/ NONE

--/

--/

pragma (page);

```
function Multiply1 (
  Left : Kernel_Time;
  Right : Hw_Long_Integer)
  return Kernel_Time;
```

```
function "" (
  Left : Kernel_Time;
  Right : Hw_Long_Integer)
  return Kernel_Time renames Multiply1;
```

```
pragma Interface(Assembly, Multiply1);
pragma Linkname(Multiply1, "GKT_multiply1");
```

```
--/ =====
--/ SUBPROGRAM NAME:
```

```
--/ ""
```

```
--/ DESCRIPTION:
```

```
--/ multiplies a Kernel_time value by an integer value returning a
```

```
--/ Kernel_time value
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ left
```

```
--/ left operand of the multiplication operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the multiplication operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the multiplication operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ takes the mathematical product of the Kernel_time value and the
```

```
--/ integer, returning a properly formatted Kernel_time result
```

```
--/ POSTCONDITIONS:
```

```
--/ the product of left and right is computed and returned
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

pragma (page);

```
--/
--/
function Multiply2 (
  Left : Hw_Long_Integer;
  Right : Kernel_Time)
  return Kernel_Time;
```

```
function "" (
  Left : Hw_Long_Integer;
  Right : Kernel_Time)
  return Kernel_Time renames Multiply2;
```

```
pragma Interface(Assembly, Multiply2);
pragma Linkname(Multiply2, "GKT_multiply2");
```

```
--/ =====
--/ SUBPROGRAM NAME:
--/ ""
```

```
--/
--/ DESCRIPTION:
--/ multiplies an Integer value by a Kernel_Time value returning a
--/ Kernel_Time value
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ left
```

```
--/ left operand of the multiplication operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the multiplication operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the multiplication operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ takes the mathematical product of the Integer and the Kernel_Time
--/ value, returning a properly formatted Kernel_Time result
```

```
--/ POSTCONDITIONS:
```

```
--/ the product of left and right is computed and returned
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

--/
--/
pregrms (page):

```
function Divide (
  Left : Kernel Time;
  Right : Hw Long Integer)
  return Kernel Time;

function "/" (
  Left : Kernel Time;
  Right : Hw Long Integer)
  return Kernel Time renames Divide;

pregrms Interface(Assembly, Divide);
pregrms Linkname(Divide, "GKT_divide");
```

--/ SUBPROGRAM NAME:

--/ "/"

--/ DESCRIPTION:

--/ divides a Kernel time value by an Integer value returning a
--/ Kernel time value

--/ --REQ: 12.1.23, 12.1.24

--/ --PRIM: N/A

--/ --TIME: N/A

--/ PARAMETERS:

--/ left

--/ left operand of the division operator (dividend)

--/ default value:

--/ none

--/ right

--/ right operand of the division operator (divisor)

--/ default value:

--/ none

--/ function return value

--/ result of the division operator

--/ PRECONDITIONS:

--/ NONE

--/ ACTIONS:

--/ takes the mathematical quotient of the Kernel time value and the
--/ integer, truncating the result toward zero if necessary, returning
--/ a properly formatted Kernel time result

--/ POSTCONDITIONS:

--/ the quotient of left divided by right is computed and returned

--/ ERROR PROCESSING:

--/ NONE

--/

--/

pragma (page);

function Unary Plus (
Left : Kernel_Time)
return Kernel_Time;

function "+" (
Left : Kernel_Time)
return Kernel_Time renames Unary_Plus;

pragma Interface(Assembly, Unary_Plus);
pragma Linkname(Unary_Plus, "GKT_unary_plus");

--/ SUBPROGRAM NAME:

--/ "+"

--/ DESCRIPTION:

--/ unary plus of a Kernel_time value returning a Kernel_time value

--/ --REQ: 12.1.23, 12.1.24

--/ --PRIM: N/A

--/ --TIME: N/A

--/ PARAMETERS:

--/ left

--/ left operand of the unary plus operator

--/ default value:

--/ none

--/ function return value

--/ result of the unary plus operator

--/ PRECONDITIONS:

--/ NONE

--/ ACTIONS:

--/ takes the mathematical unary plus of the Kernel_time value, returning

--/ a properly formatted Kernel_time result

--/ POSTCONDITIONS:

--/ the unary plus of left is computed and returned

--/ ERROR PROCESSING:

--/ NONE

pragma (page);

```

function Unary_Minus (
  Left : Kernel_Time)
  return Kernel_Time;

function "*" (
  Left : Kernel_Time)
  return Kernel_Time renames Unary_Minus;

pragma Interface(Assembly, Unary_Minus);
pragma Linkname(Unary_Minus, "GKT_unary_minus");

--/ SUBPROGRAM NAME:
--/ "*"
--/ DESCRIPTION:
--/ unary negation of a Kernel_time value returning a Kernel_time value
--/ --REQ: 12.1.23, 12.1.24
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ left operand of the unary negation operator
--/ default value:
--/ none
--/ function return value
--/ result of the negation operator
--/ PRECONDITIONS:
--/ NONE
--/ ACTIONS:
--/ takes the mathematical unary negation of the Kernel_time value,
--/ returning a properly formatted Kernel_time result
--/ POSTCONDITIONS:
--/ the unary negation of left is computed and returned
--/ ERROR PROCESSING:
--/ NONE
--/
pragma (page);

```

```

function Ls (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Boolean;

function "<" (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Boolean renames Ls;

pragma Interface(Assembly, Ls);
pragma Linkname(Ls, "GKT_ls");

--/ SUBPROGRAM NAME:
--/ "<"
--/ DESCRIPTION:
--/ compares two Kernel_time values returning a Boolean true if left is
--/ less than right, false otherwise
--/ --REQ: 12.1.23, 12.1.24
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ left
--/ left operand of the less-than operator
--/ default value:
--/ none
--/ right
--/ right operand of the less-than operator
--/ default value:
--/ none
--/ function return value
--/ result of the less-than operator
--/ PRECONDITIONS:
--/ NONE
--/ ACTIONS:
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is less than the right, returning false otherwise
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/ ERROR PROCESSING:
--/ NONE

```

```

--/
--/ EXAMPLES:
--/ <sample calls and explanations>
--/

```

```

pragma (page);

```

```

function Le (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Boolean;

function "<=" (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Boolean renames Le;

pragma Interface(Assembly, Le);
pragma Linkname(Le, "GKT_le");

--/ =====
--/ SUBPROGRAM NAME:
--/ "<="
--/
--/ DESCRIPTION:
--/ compares two Kernel_time values returning a Boolean true if left is
--/ less than or equal to right, false otherwise
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the less-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the less-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the less-than-or-equal-to operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is less than or equal to the right, returning false
--/ otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:

```


--/ NONE

--/

--/

pragma (page);

function Gr (
 Left : Kernel_Time;
 Right : Kernel_Time)
 return Boolean;

function ">" (
 Left : Kernel_Time;
 Right : Kernel_Time)
 return Boolean renames Gr;

pragma Interface(Assembly, Gr);
pragma LinkName(Gr, "GKT_gr");

--/ SUBPROGRAM NAME:
--/ ">"

--/ DESCRIPTION:

--/ compares two Kernel_time values returning a Boolean true if left is
--/ greater than right, false otherwise

--/ --REQ: 12.1.23, 12.1.24

--/ --PRIM: N/A

--/ --TIME: N/A

--/ PARAMETERS:

--/ left

--/ left operand of the greater-than operator

--/ default value:

--/ none

--/ right

--/ right operand of the greater-than operator

--/ default value:

--/ none

--/ function return value

--/ result of the greater-than operator

--/ PRECONDITIONS:

--/ NONE

--/ ACTIONS:

--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is greater than the right, returning false otherwise

--/ POSTCONDITIONS:

--/ the values of left and right are compared

--/ ERROR PROCESSING:

--/ NONE

```

--// EXAMPLES:
--// <sample calls and explanations>
--//

```

```

pragma (page);

```

```

function Ge (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Boolean;

function ">=" (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Boolean renames Ge;

pragma Interface(Assembly, Ge);
pragma LinkName(Ge, "GKT_g");

```

```

--// SUBPROGRAM NAME:
--// ">="

```

```

--// DESCRIPTION:
--// compares two Kernel_time values returning a Boolean true if left is
--// greater than or equal to right, false otherwise

```

```

--// --REQ: 12.1.23, 12.1.24

```

```

--// --PRIM: N/A

```

```

--// --TIME: N/A

```

```

--// PARAMETERS:

```

```

--// left

```

```

--// left operand of the greater-than-or-equal to operator

```

```

--// default value:

```

```

--// none

```

```

--// right

```

```

--// right operand of the greater-than-or-equal to operator

```

```

--// default value:

```

```

--// none

```

```

--// function return value

```

```

--// result of the greater-than-or-equal to operator

```

```

--// PRECONDITIONS:

```

```

--// NONE

```

```

--// ACTIONS:

```

```

--// compares the two Kernel_time values, returning Boolean true if the

```

```

--// left operand is greater than or equal to the right; returning

```

```

--// false otherwise

```

```

--// POSTCONDITIONS:

```

```

--// the values of left and right are compared

```

```

--// ERROR PROCESSING:

```

--/ NONE

--/

--/

end Generic_Kernel_Time;
pragma (page);

```
--/ *****
--/ MODULE NAME:
--/   Kernel_time
--/
--/ MODULE TYPE:
--/   package specification
--/   package body
--/
--/ MODULE PURPOSE:
--/   an example of (and a template for) an instantiation of the generic
--/   package generic_Kernel_time
--/
--/ *****
--/ MODULE DESCRIPTION:
--/   package instantiating the Kernel_time abstract type and subprograms to
--/   manipulate it where:
--/   ticks_per_second_value is defined as 16,384
--/
--/ MODULE CONTENTS:
--/   generic Kernel_time (generic package specification)
--/   integral duration (type)
--/   Kernel_time (private type)
--/   seconds (overloaded (2) function)
--/   ticks_per_second (constant)
--/   "+" (function)
--/   "*" (function)
--/   "**" (overloaded (2) function)
--/   "/" (function)
--/   generic Kernel_time (generic package body)
--/
--/ NOTES:
--/   this package can be instantiated and used in place of the
--/   compiler-provided Package_Calendar
--/
--/ *****
--/ MODIFICATION HISTORY:
--/   18aug88  bamberg  created (complies clean)
--/   11jan89  lirth  final doc. update for Beta Release (no change)
--/
--/ *****
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/   The following copyright must be included in this software and
--/   all software utilizing this software.
--/
--/ DISTRIBUTION:
--/   Approved for public release, distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
```

.../ Department of Defense under contract F19628-85-C-0003, the SEI .../ is supported by the services and defense agencies, with the U.S. .../ Air Force as the executive contracting agent.

Permission to use, copy, modify, or distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Further, the names of Software Engineering Institute or Carnegie Mellon University may not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. CMU makes no claims or representations about the suitability of this software for any purpose. This software is provided "as is" and no warranty, express or implied, is made by the SEI or CMU, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty. No responsibility is assumed by the SEI or CMU in connection herewith.

```
with Generic_Kernel_Time;
pragma Elaborate (Generic_Kernel_Time);
-- [o cat: see MODULE CONTENTS
```

.../package short name: KT

Package Kernel Time

```
Is new Generic Kernel Time (
  Ticks Per Second_Value => 500_000);
pragma (page);
```

[illegible]

```

./ .....
./ MODULE NAME:
./   generic_time_globals
./
./ MODULE TYPE:
./   generic_package_specification
./
./ MODULE PURPOSE:
./   exports the type "time" as an alternative to an application's having
./   to use package calendar and its type "time".
./ .....
./ MODULE DESCRIPTION:
./   exports type time in two distinct flavors:
./   epoch_time (conceptually related to Ada's type TIME); and
./   elapsed_time (conceptually related to Ada's type DURATION)
./   and the "expected" operations to preserve those two abstractions
./ .....
./ MODULE CONTENTS:
./   generic_time_globals (package specification)
./   base_time (function)
./   create_elapsed_time (function)
./   create_epoch_time (function)
./   elapsed_time (private type)
./   integral_duration (type)
./   milliseconds (function)
./   microseconds (function)
./   seconds (overloaded (2) function)
./   ticks_per_second (constant)
./   to_Ada_duration (function)
./   to_elapsed_time (overloaded (2) function)
./   to_epoch_time (function)
./   to_kernel_time (overloaded (2) function)
./   zero_elapsed_time (constant)
./   zero_epoch_time (constant)
./   "+" (overloaded (3) function)
./   "*" (overloaded (2) function)
./   "/" (function)
./   "<" (overloaded (2) function)
./   "<=" (overloaded (2) function)
./   ">" (overloaded (2) function)
./   ">=" (overloaded (2) function)
./
./ REFERENCES
./   DESIGN DOCUMENTS:
./   Kernel Facilities Definition Document
./   --REQ: 12.1.1, 12.1.25
./   --PRIM: N/A
./   --TIME: N/A
./
./ USER'S MANUAL:
./   Kernel User's Manual
./

```

```

./ TESTING AND VALIDATION:
./   Tested via unit tests for generic_kernel_time
./
./ NOTES:
./   none
./ .....
./ MODIFICATION HISTORY:
./   18aug88 bambberg created (complies clean)
./   4jan89 firh revised; added more functions following
./   13jan89 firh model in Kernel_time
./   17jan89 firh final check for beta release
./   minor format changes
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./
./ The following copyright must be included in this software and
./ all software utilizing this software.
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....
./ pragma (page);

```

```

:= Kernel_Time.Ticks Per Second;

pragma (page);

```

```

with Hardware_Interface; use Hardware_Interface;
.. to get: hw_duration
.. hw_integer
.. hw_long_integer

with Kernel_Time;
.. to get: integral_duration
.. Kernel_time
.. zero Kernel time
.. ticks_per_second

generic
.. / .....
.. / package short name: GIG
.. / .....

package Generic_Time_Globals
is
..
.. elapsed_time conceptually corresponds to the Ada type "duration"
..
type Elapsed_Time is private;
..
.. there is a zero value for elapsed_time
..
Zero_Elapsed_Time : constant Elapsed_Time;
..
.. epoch time conceptually corresponds to the Ada type "time"
..
type Epoch_Time is private;
..
.. there is a zero value for epoch_time
..
Zero_Epoch_Time : constant Epoch_Time;
..
.. an integral duration is an integer-valued measure of duration in
.. user-specified units
..
type Integral_Duration is new Kernel_Time.Integral_Duration;
.. the value ticks_per_second is reexported for completeness' sake
Ticks_Per_Second : constant Hw_Long_Integer

```

```

function Create Elapsed Time (
Day : Hw Long Integer;
Second : Hw Duration)
return Elapsed Time;
--/
--/ SUBPROGRAM NAME:
--/ create_elapsed_time
--/
--/ DESCRIPTION:
--/ create an elapsed time value from its component parts
--/
--/ --REQ: 12.1.9
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ day any integer number of days
--/
--/ default value:
--/ none
--/
--/ second
--/ number of seconds within that day (note that this limits
--/ legal values to 0 to 86400 seconds, 1 day)
--/
--/ function return value
--/ an elapsed time value created from its component parts
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the specified time in elapsed_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
pragma (page);

```

```

function Create Epoch Time (
Day : Hw Long Integer;
Second : Hw Duration)
return Epoch Time;
--/
--/ SUBPROGRAM NAME:
--/ create_epoch_time
--/
--/ DESCRIPTION:
--/ create an epoch time value from its component parts
--/
--/ --REQ: 12.1.9
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ day any integer number of days
--/
--/ default value:
--/ none
--/
--/ second
--/ number of seconds within that day (note that this limits
--/ legal values to 0 to 86400 seconds, 1 day)
--/
--/ function return value
--/ an epoch time value created from its component parts
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the specified time in epoch_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
pragma (page);

```

```

function Seconds (
  Ada_Duration : Hw_Duration)
  return Elapsed_Time;
--/
--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Ada duration type to elapsed_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Ada duration
--/ the time to convert in Ada duration format (note that this limits
--/ legal values to 0 to 86400 seconds, 1 day)
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input duration in elapsed_time format (note that this contains
--/ time up to 1 day)
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the Ada duration time in elapsed_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```

```

function Seconds (
  An_Integral_Duration : Integral_Duration)
  return Elapsed_Time;
--/
--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed time to elapsed_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ an integral duration
--/ the time to convert in integer format
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input integral duration in elapsed_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the Integral duration time in elapsed_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```



```
function Milliseconds (
Integral_Milliseconds : Integral_Duration)
return Elapsed_Time;
```

```
--/ SUBPROGRAM NAME:
```

```
--/ milliseconds
```

```
--/ DESCRIPTION:
```

```
--/ conversion routine from integral elapsed time to elapsed_time
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ Integral milliseconds
```

```
--/ the duration in milliseconds to convert to elapsed time
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ the input integral duration in elapsed_time format
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ returns the integral duration time in elapsed_time format
```

```
--/ POSTCONDITIONS:
```

```
--/ NONE
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

```
--/ pragma (page);
```

```
function Microseconds (
Integral_Microseconds : Integral_Duration)
return Elapsed_Time;
```

```
--/ SUBPROGRAM NAME:
```

```
--/ microseconds
```

```
--/ DESCRIPTION:
```

```
--/ conversion routine from integral elapsed time to elapsed_time
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ Integral microseconds
```

```
--/ the duration in microseconds to convert to elapsed time
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ the input integral duration in elapsed_time format
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ returns the integral duration time in elapsed_time format
```

```
--/ POSTCONDITIONS:
```

```
--/ NONE
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

```
--/ pragma (page);
```

```

function "+" (
  Left : Epoch Time;
  Right : Elapsed Time)
return Epoch Time;
--
-- SUBPROGRAM NAME:
--
--
-- DESCRIPTION:
-- subtracts the right elapsed time value from the left epoch time value
-- returning an epoch time value
--
-- REQ: 12 1 23 12 1 24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left left operand of the subtraction operator (subtrahend)
-- default value:
-- none
--
-- right right operand of the subtraction operator (minuend)
-- default value:
-- none
--
-- function return value
-- result of the subtraction operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical difference of the two values by subtracting
-- the right operand from the left operand, returning a properly
-- formatted epoch time result
--
-- POSTCONDITIONS:
-- the difference of left and right is computed and returned
--
-- ERROR PROCESSING:
-- NONE
--
--
pragma (page);

```

```

function "*" (
  Left : Epoch Time;
  Right : Elapsed Time)
return Epoch Time;
--
-- SUBPROGRAM NAME:
--
--
-- DESCRIPTION:
-- adds an epoch time value and an elapsed time value returning an
-- epoch time value
--
-- REQ: 12 1 23 12 1 24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left left operand of the addition operator
-- default value:
-- none
--
-- right right operand of the addition operator
-- default value:
-- none
--
-- function return value
-- result of the addition operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical sum of the epoch time value and the
-- elapsed time value, returning a properly formatted epoch time
-- result
--
-- POSTCONDITIONS:
-- the sum of left and right is computed and returned
--
-- ERROR PROCESSING:
-- NONE
--
--
pragma (page);

```

```

function "-" (
    Left: Epoch Time;
    Right: Epoch Time;
    return Elapsed Time;
    //
    // SUBPROGRAM NAME:
    //
    // DESCRIPTION:
    // subtracts the right epoch time value from the left epoch time value
    // returning an elapsed time value
    //
    // --REQ: 12.1.23, 12.1.24
    //
    // --PRIM: N/A
    //
    // --TIME: N/A
    //
    // PARAMETERS:
    // left
    // left operand of the subtraction operator (subtrahend)
    //
    // default value:
    // none
    //
    // right
    // right operand of the subtraction operator (minuend)
    //
    // default value:
    // none
    //
    // function return value
    // result of the subtraction operator
    //
    // PRECONDITIONS:
    // NONE
    //
    // ACTIONS:
    // takes the mathematical difference of the two values by subtracting
    // the right operand from the left operand, returning a properly
    // formatted elapsed_time result
    //
    // POSTCONDITIONS:
    // the difference of left and right is computed and returned
    //
    // ERROR PROCESSING:
    // NONE
    //
    //
    // pragma (page);

```

```

function Base Time return Epoch Time;
//
// SUBPROGRAM NAME:
// base time
//
// DESCRIPTION:
// returns the value of the base time that was set during processor
// initialization
//
// --REQ: 12.1.23, 12.1.24
//
// --PRIM: N/A
//
// --TIME: N/A
//
// PARAMETERS:
// NONE
//
// PRECONDITIONS:
// NONE
//
// ACTIONS:
// returns the value of the base time
//
// POSTCONDITIONS:
// NONE
//
// ERROR PROCESSING:
// NONE
//
//
// pragma (page);

```

```

function "+" (
  Left : Elapsed Time;
  Right : Elapsed Time)
return Elapsed Time;

```

```

--/
--/ SUBPROGRAM NAME:

```

```

--/ DESCRIPTION:

```

```

--/ adds two elapsed_time values returning an elapsed_time value

```

```

--/ --REQ: 12.1.23, 12.1.24

```

```

--/ --PRIM: N/A

```

```

--/ --TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ left left operand of the addition operator

```

```

--/ default value:

```

```

--/ none

```

```

--/ right

```

```

--/ right operand of the addition operator

```

```

--/ default value:

```

```

--/ none

```

```

--/ function return value

```

```

--/ result of the addition operator

```

```

--/ PRECONDITIONS:

```

```

--/ NONE

```

```

--/ ACTIONS:

```

```

--/ takes the mathematical sum of the two elapsed_time values, returning

```

```

--/ a properly formatted elapsed_time result

```

```

--/ POSTCONDITIONS:

```

```

--/ the sum of left and right is computed and returned

```

```

--/ ERROR PROCESSING:

```

```

--/ NONE

```

```

pragma (page);

```

```

function "-" (
  Left : Elapsed Time;
  Right : Elapsed Time)
return Elapsed Time;

```

```

--/
--/ SUBPROGRAM NAME:

```

```

--/ DESCRIPTION:

```

```

--/ subtracts the right elapsed_time value from the left elapsed_time

```

```

--/ value returning an elapsed_time value

```

```

--/ --REQ: 12.1.23, 12.1.24

```

```

--/ --PRIM: N/A

```

```

--/ --TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ left left operand of the subtraction operator (subtrahend)

```

```

--/ default value:

```

```

--/ none

```

```

--/ right

```

```

--/ right operand of the subtraction operator (minuend)

```

```

--/ default value:

```

```

--/ none

```

```

--/ function return value

```

```

--/ result of the subtraction operator

```

```

--/ PRECONDITIONS:

```

```

--/ NONE

```

```

--/ ACTIONS:

```

```

--/ takes the mathematical difference of the two elapsed_time values by

```

```

--/ subtracting the right operand from the left operand, returning a

```

```

--/ properly formatted elapsed_time result

```

```

--/ POSTCONDITIONS:

```

```

--/ the difference of left and right is computed and returned

```

```

--/ ERROR PROCESSING:

```

```

--/ NONE

```

```

pragma (page);

```

```

function *** (
  Left : Elapsed Time;
  Right : Hw Integer)
return Elapsed Time;
--/
--/ SUBPROGRAM NAME:
--/
--/
--/ DESCRIPTION:
--/ multiplies an elapsed_time value by an integer value returning an
--/ elapsed_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the multiplication operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the multiplication operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the multiplication operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical product of the elapsed_time value and the
--/ integer, returning a properly formatted elapsed_time result
--/
--/ POSTCONDITIONS:
--/ the product of left and right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```

```

function *** (
  Left : Hw Integer;
  Right : Elapsed Time)
return Elapsed Time;
--/
--/ SUBPROGRAM NAME:
--/
--/
--/ DESCRIPTION:
--/ multiplies an integer value by an elapsed_time value returning an
--/ elapsed_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the multiplication operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the multiplication operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the multiplication operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical product of the integer and the elapsed_time
--/ value, returning a properly formatted elapsed_time result
--/
--/ POSTCONDITIONS:
--/ the product of left and right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```

```

function "/" (
    Left : Elapsed_Time;
    Right : Intw_Integer)
return Elapsed_Time;
--/
--/ SUBPROGRAM NAME:
--/ "/"
--/
--/ DESCRIPTION:
--/ divides an elapsed_time value by an integer value returning an
--/ elapsed_time value
--/
--/ --REQ: 12.1.23.12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left left operand of the division operator (dividend)
--/
--/ default value:
--/ none
--/
--/ right right operand of the division operator (divisor)
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the division operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical quotient of the elapsed_time value and the
--/ integer, returning a properly formatted elapsed_time result
--/
--/ POSTCONDITIONS:
--/ the quotient of left divided by right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```

```

function "<" (
    Left : Epoch_Time;
    Right : Epoch_Time)
return Boolean;
--/
--/ SUBPROGRAM NAME:
--/ "<"
--/
--/ DESCRIPTION:
--/ compares two epoch_time values returning a Boolean true if left is
--/ less than right, false otherwise
--/
--/ --REQ: 12.1.23.12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left left operand of the less-than operator
--/
--/ default value:
--/ none
--/
--/ right right operand of the less-than operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the less-than operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two epoch_time values, returning Boolean true if the
--/ left operand is less than the right; returning false otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/
--/ pragma (page);

```

```
function "<=" {
  Left : Epoch Time;
  Right : Epoch Time;
  return Boolean;
}
```

```
--/ SUBPROGRAM NAME:
--/ "<="
```

```
--/ DESCRIPTION:
```

```
--/ compares two epoch time values returning a Boolean true if left is
--/ less than or equal to right, false otherwise
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ left
```

```
--/ left operand of the less-than-or-equal-to operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the less-than-or-equal-to operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the less-than-or-equal-to operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ compares the two epoch time values, returning Boolean true if the
--/ left operand is less than or equal to the right, returning false
--/ otherwise
```

```
--/ POSTCONDITIONS:
--/ the values of left and right are compared
```

```
--/ ERROR PROCESSING:
--/ NONE
```

```
--/ pragma (page):
```

```
function ">" {
  Left : Epoch Time;
  Right : Epoch Time;
  return Boolean;
}
```

```
--/ SUBPROGRAM NAME:
--/ ">"
```

```
--/ DESCRIPTION:
```

```
--/ compares two epoch time values returning a Boolean true if left is
--/ greater than right, false otherwise
```

```
--/ --REQ: 12.1.23, 12.1.24
```

```
--/ --PRIM: N/A
```

```
--/ --TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ left
```

```
--/ left operand of the greater-than operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the greater-than operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the greater-than operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ compares the two epoch time values, returning Boolean true if the
--/ left operand is greater than the right, returning false otherwise
```

```
--/ POSTCONDITIONS:
--/ the values of left and right are compared
```

```
--/ ERROR PROCESSING:
--/ NONE
```

```
--/ pragma (page):
```

```

function ">=" (
  Left : Epoch Time;
  Right : Epoch Time)
return Boolean;

```

```

--/ SUBPROGRAM NAME:
--/ ">="

```

```

--/ DESCRIPTION:
--/ compares two epoch time values returning a Boolean true if left is
--/ greater than or equal to right, false otherwise

```

```

--/ REQ: 12.1.23, 12.1.24

```

```

--/ PRIM: N/A

```

```

--/ TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ left
--/ left operand of the greater-than-or-equal-to operator

```

```

--/ default value:
--/ none

```

```

--/ right
--/ right operand of the greater-than-or-equal-to operator

```

```

--/ default value:
--/ none

```

```

--/ function return value
--/ result of the greater-than-or-equal-to operator

```

```

--/ PRECONDITIONS:
--/ NONE

```

```

--/ ACTIONS:
--/ compares the two epoch time values, returning Boolean true if the
--/ left operand is greater than or equal to the right, returning
--/ false otherwise

```

```

--/ POSTCONDITIONS:
--/ the values of left and right are compared

```

```

--/ ERROR PROCESSING:
--/ NONE

```

```

pragma (page);

```

```

function "<" (
  Left : Elapsed Time;
  Right : Elapsed Time)
return Boolean;

```

```

--/ SUBPROGRAM NAME:
--/ "<"

```

```

--/ DESCRIPTION:
--/ compares two elapsed time values returning a Boolean true if left is
--/ less than right, false otherwise

```

```

--/ REQ: 12.1.23, 12.1.24

```

```

--/ PRIM: N/A

```

```

--/ TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ left
--/ left operand of the less-than operator

```

```

--/ default value:
--/ none

```

```

--/ right
--/ right operand of the less-than operator

```

```

--/ default value:
--/ none

```

```

--/ function return value
--/ result of the less-than operator

```

```

--/ PRECONDITIONS:
--/ NONE

```

```

--/ ACTIONS:
--/ compares the two elapsed time values, returning Boolean true if the
--/ left operand is less than the right, returning false otherwise

```

```

--/ POSTCONDITIONS:
--/ the values of left and right are compared

```

```

--/ ERROR PROCESSING:
--/ NONE

```

```

pragma (page);

```



```

function "<" (
  Left : Elapsed Time;
  Right : Elapsed Time)
return Boolean;

-- SUBPROGRAM NAME:
-- "<"
--
-- DESCRIPTION:
-- compares two elapsed time values returning a Boolean true if left is
-- less than or equal to right, false otherwise
--
-- REQ: 12.1.23, 12.1.24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the less-than-or-equal-to operator
--
-- default value:
-- none
--
-- right
-- right operand of the less-than-or-equal-to operator
--
-- default value:
-- none
--
-- function return value
-- result of the less-than-or-equal-to operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- compares the two elapsed time values, returning Boolean true if the
-- left operand is less than or equal to the right, returning false
-- otherwise
--
-- POSTCONDITIONS:
-- the values of left and right are compared
--
-- ERROR PROCESSING:
-- NONE
--
-- pragma (page);

function ">" (
  Left : Elapsed Time;
  Right : Elapsed Time)
return Boolean;

-- SUBPROGRAM NAME:
-- ">"
--
-- DESCRIPTION:
-- compares two elapsed time values returning a Boolean true if left is
-- greater than right, false otherwise
--
-- REQ: 12.1.23, 12.1.24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the greater-than operator
--
-- default value:
-- none
--
-- right
-- right operand of the greater-than operator
--
-- default value:
-- none
--
-- function return value
-- result of the greater-than operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- compares the two elapsed time values, returning Boolean true if the
-- left operand is greater than the right, returning false otherwise
--
-- POSTCONDITIONS:
-- the values of left and right are compared
--
-- ERROR PROCESSING:
-- NONE
--
-- pragma (page);

```

```

function ">=" (
  Left : Elapsed_Time;
  Right : Elapsed_Time)
return Boolean;

```

```

-- SUBPROGRAM NAME:
-- ">="

```

```

-- DESCRIPTION:
-- compares two elapsed_time values returning a Boolean true if left is
-- greater than or equal to right, false otherwise

```

```

-- REQ: 12.1.23, 12.1.24

```

```

-- PRIM: N/A

```

```

-- TIME: N/A

```

```

-- PARAMETERS:

```

```

-- left
-- left operand of the greater-than-or-equal-to operator

```

```

-- default value:
-- none

```

```

-- right
-- right operand of the greater-than-or-equal-to operator

```

```

-- default value:
-- none

```

```

-- function return value
-- result of the greater-than-or-equal-to operator

```

```

-- PRECONDITIONS:
-- NONE

```

```

-- ACTIONS:

```

```

-- compares the two elapsed_time values, returning Boolean true if the
-- left operand is greater than or equal to the right, returning
-- false otherwise

```

```

-- POST CONDITIONS:

```

```

-- the values of left and right are compared

```

```

-- ERROR PROCESSING:
-- NONE

```

```

pragma (page);

```

```

function To_Elapsed_Time (
  An_Ada_Duration : Hw_Duration)
return Elapsed_Time;
-- SUBPROGRAM NAME:
-- to_elapsed_time

```

```

-- DESCRIPTION:
-- converts values of Ada type duration to elapsed_time values

```

```

-- REQ: 12.1.23, 12.1.24

```

```

-- PRIM: N/A

```

```

-- TIME: N/A

```

```

-- PARAMETERS:

```

```

-- an_Ada_duration
-- a value of a duration type

```

```

-- default value:
-- none

```

```

-- function return value
-- result of the conversion

```

```

-- PRECONDITIONS:
-- NONE

```

```

-- ACTIONS:

```

```

-- returns the value as a properly formatted elapsed_time value

```

```

-- POSTCONDITIONS:
-- NONE

```

```

-- ERROR PROCESSING:
-- NONE

```

```

pragma (page);

```

```

function To_Ada_Duration (
  An_Elapsed_Time : Elapsed_Time)
return Ada_Duration;
--/
--/ SUBPROGRAM NAME:
--/ to_Ada_duration
--/
--/ DESCRIPTION:
--/ converts values of Kernel type elapsed_time to Ada type duration
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ an elapsed_time
--/ a value of type elapsed_time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the conversion
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the value as an Ada duration type value
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ when the result of the conversion would cause overflow:
--/ the Ada predefined exception CONSTRAINT_ERROR is raised
--/ POSTCONDITIONS:
--/ the conversion is not performed
--/
--/ pragma (page);

```

```

function To_Kernel_Time (
  Elapsed_Time : Elapsed_Time)
return Kernel_Time;
--/
--/ SUBPROGRAM NAME:
--/ to_Kernel_time
--/
--/ DESCRIPTION:
--/ *** INTERNAL KERNEL USE ONLY ***
--/ in keeping with maintenance of elapsed_time as an abstract,
--/ private type, this conversion operator is needed
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ elapsed
--/ the time as an elapsed_time value
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the time as a Kernel_time value
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ none; just re-types the elapsed_time value as a Kernel_time value
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/ pragma (page);

```

```

function To_Kernel_Time (
    Epoch : Epoch_Time)
return Kernel_Time_Kernel_Time;
--/ SUBPROGRAM NAME:
--/ to_Kernel_time
--/
--/ DESCRIPTION:
--/ *** INTERNAL KERNEL USE ONLY ***
--/ In keeping with maintenance of epoch_time as an abstract,
--/ private type, this conversion operator is needed
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ elapsed
--/ the time as an epoch_time value
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the time as a Kernel_time value
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ none, just re-types the epoch_time value as a Kernel_time value
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/ pragma (page);

```

```

function To_Elapsed_Time (
    Ktime : Kernel_Time_Kernel_Time)
return Elapsed_Time;
--/ SUBPROGRAM NAME:
--/ to_elapsed_time
--/
--/ DESCRIPTION:
--/ *** INTERNAL KERNEL USE ONLY ***
--/ In keeping with maintenance of elapsed_time as an abstract,
--/ private type, this conversion operator is needed
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Ktime
--/ the time as an Internal_Kernel_time value
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the time as elapsed_time value
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ none, just re-types the Kernel_time value as an elapsed_time value
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/ pragma (page);

```

```

function To_Epoch_Time (
  KTime : Kernel_Time_Kernel_Time)
return Epoch_Time;
--/
--/
--/ SUBPROGRAM NAME:
--/ to_epoch_time
--/
--/ DESCRIPTION:
--/ *** INTERNAL KERNEL USE ONLY ***
--/ In keeping with maintenance of epoch_time as an abstract,
--/ private type, this conversion operator is needed
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ KTime
--/ the time as an internal Kernel_time value
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the time as epoch_time value
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ none, just re-types the Kernel_time value as an epoch_time value
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/
--/
pragma (page);

```

```

--/
--/
private
--/
--/ elapsed_time is based on Kernel_time
--/
--/
type Elapsed_Time is new Kernel_Time_Kernel_Time;
--/
--/ the zero value is also based on Kernel_time
--/
--/
Zero_Elapsed_Time : constant Elapsed_Time :=
  Elapsed_Time (Kernel_Time_Zero_Kernel_Time);
--/
--/ epoch_time is based on Kernel_time
--/
--/
type Epoch_Time is new Kernel_Time_Kernel_Time;
--/
--/ the zero value is also based on Kernel_time
--/
--/
Zero_Epoch_Time : constant Epoch_Time :=
  Epoch_Time (Kernel_Time_Zero_Kernel_Time);
--/
end Generic_Time_Globals;
pragma (page);

```

```

.....
--/ MODULE NAME:
--/   time_globals
--/
--/ MODULE TYPE:
--/   package specification
--/
--/ MODULE PURPOSE:
--/   an example of (and a template for) an instantiation of the generic
--/   package generic_time_globals
--/
--/
--/
--/ MODU: E DESCRIPTION:
--/   package instantiating the type time in two distinct flavors:
--/   epoch_time (conceptually related to Ada's type TIME), and
--/   elapsed_time (conceptually related to Ada's type DURATION)
--/   and the "expected" operations to preserve those two abstractions
--/
--/
--/ MODULE CONTENTS:
--/   time_globals (package specification)
--/   base_time (function)
--/   create_elapsed_time (function)
--/   create_epoch_time (function)
--/   elapsed_time (private type)
--/   epoch_time (private type)
--/   integral_duration (type)
--/   milliseconds (function)
--/   microseconds (function)
--/   seconds (overloaded (2) function)
--/   ticks_per_second (constant)
--/   to_Ada_duration (function)
--/   to_elapsed_time (overloaded (2) function)
--/   to_epoch_time (function)
--/   to_Kernel_time (overloaded (2) function)
--/   zero_elapsed_time (constant)
--/   zero_epoch_time (constant)
--/   "*" (overloaded (2) function)
--/   "+" (overloaded (2) function)
--/   "-" (overloaded (2) function)
--/   "<" (overloaded (2) function)
--/   "<=" (overloaded (2) function)
--/   ">" (overloaded (2) function)
--/   ">=" (overloaded (2) function)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS
--/     Kernel Facilities Definition
--/
--/   USER'S MANUAL
--/     Kernel User's Manual
--/
--/   TESTING AND VALIDATION
--/     none
--/
--/ NOTES:
--/   none
--/
--/
--/ MODIFICATION HISTORY:
--/   18Aug88   bamberg   created (complies clean)
--/   6Jan89   firh       revised comments after changes to generic_dito
--/   13Jan89   firh       final check for beta release
--/
--/
--/ VERSION: 3.0
--/
--/
--/ DISCLAIMER:
--/   The following copyright must be included in this software and
--/   all software utilizing this software.
--/
--/ DISTRIBUTION:
--/   Approved for public release; distribution is unlimited.
--/
--/   Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/   PA. The Software Engineering Institute (SEI) is a federally
--/   funded research and development center established and operated
--/   by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/   Department of Defense under contract F19628-85-C-0003, the SEI
--/   is supported by the services and defense agencies, with the U.S.
--/   Air Force as the executive contracting agent.
--/
--/   Permission to use, copy, modify, or distribute this software and
--/   its documentation for any purpose and without fee is hereby
--/   granted, provided that the above copyright notice appear in all
--/   copies and that both that copyright notice and this permission
--/   notice appear in supporting documentation. Further, the names
--/   of Software Engineering Institute or Carnegie Mellon University may
--/   not be used in advertising or publicity pertaining to
--/   distribution of the software without specific, written prior
--/   permission. CMU makes no claims or representations about the
--/   suitability of this software for any purpose. This software is
--/   provided "as is" and no warranty, express or implied, is made
--/   by the SEI or CMU, as to the accuracy and functioning of the
--/   program and related program material, nor shall the fact of
--/   distribution constitute any such warranty. No responsibility is
--/   assumed by the SEI or CMU in connection herewith.
--/
--/
--/ with Generic Time Globals;
--/ pragma Elaborate (Generic Time Globals);
--/ .. to get: see MODULE CONTENTS
--/
--/ .....
--/ package short name: TG
--/ .....
--/
--/ package Time Globals

```

is new Generic_Time_Globals;
pragma (page);

```

.....
--/ MODULE NAME:
--/   generic_schedule_types
--/
--/ MODULE TYPE:
--/   generic_package_specification
--/   generic_package_body
--/
--/ MODULE PURPOSE:
--/   priority_abstract_type_declaration
--/   preemption_type_declaration
--/   process_state_type_declaration
--/
--/
--/ MODULE DESCRIPTION:
--/   provides the abstract type declaration for scheduling attributes:
--/   priority, preemption, and process state
--/
--/ MODULE CONTENTS:
--/   generic_schedule_types (generic_package_specification)
--/   current_process_priority (constant)
--/   default_preemption (constant)
--/   default_priority (constant)
--/   default_process_state (constant)
--/   highest_priority (constant)
--/   lowest_priority (constant)
--/   lowest_priority_value (generic_formal_parameter)
--/   null_priority_range (exception)
--/   null_priority_range_enabled (generic_formal_parameter)
--/   preemption (type)
--/   priority (type)
--/   process_state (type)
--/   generic_schedule_types (generic_package_body)
--/   elaboration_time_initialization_code
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     Kernel_Facilities_Definition
--/     --REQ: 5.1.6: 5.1.8;
--/     --REQ: 9.1.2; 9.1.3; 9.1.4; 9.1.5; 9.1.10; 9.1.23
--/     --PRIM: N/A
--/     --TIME: N/A
--/
--/ USER'S MANUAL:
--/   Kernel_User's_Manual
--/
--/ TESTING AND VALIDATION:
--/   none
--/
--/ NOTES:
--/   none
--/
--/
--/ MODIFICATION HISTORY:
--/   18aug88 bamborg created (complies clean)

```

```

../ 01sep88 r/s      modified to export the value ticks per slice
../ 07oct88 bamberg  reorder priority declarations and get type of
../               lowest priority correct!
../ 12dec88 r/s      modify legal priority range to allow for a
../               background with the lowest possible priority
../               (i.e. one that won't interfere with user
../               processes)
../ 13dec88 dvk      deleted maximum slices per quantum and
../               ticks per slice. This functionality has been
../               moved to generic timeslice management.
../ 15feb89 bamberg  update commentary on priority per bug report
../               0006_SEI
../ 27aug89 bamberg  final doc. check for 01sep89 freeze
../
../ -----
../ VERSION: 3.0
../
../ DISCLAIMER:
../
../ The following copyright must be included in this software and
../ all software utilizing this software.
../
../ DISTRIBUTION:
../
../ Approved for public release; distribution is unlimited
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ of Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../

```

with Kernel Exceptions;

```
.. to get: null priority range
```

with Hardware Interface, use Hardware Interface;

```

.. to get: hw Integer
..           hw natural
..           hw positive

generic:

.. ERROR checking conditional compilation flags
..
.. REQ: 5.1.8
..
Null Priority Range Enabled: In Boolean := True;

.. value indicating the minimum possible priority value (this value is
.. never used within the Kernel except for the initialization of the
.. exported constant and the lower bound on priority, nor should it be used
.. without the Kernel)
..
.. REQ: 5.1.6; 9.1.4; 9.1.5
..
Lowest Priority Value: In Hw_Positive;
..
.. package short name: GST
..
package Generic_Schedule_Types
is
Null Priority Range : exception
renames Kernel_Exceptions Null Priority Range;

.. KERNEL PRIORITY MODEL:
..
.. the highest priority that should be used by an application process is 1;
.. this is available to the application via the constant highest_priority
.. the lowest priority that should be used by an application process is
.. specified by the tailoring parameter lowest_priority_value;
.. this is available to the application via the constant lowest_priority
.. the Kernel reserves priority'first (0) and
.. priority'last (lowest_priority_value + 1) for its own use.
..
.. thus, the application should ONLY use priorities in the range:
.. highest_priority .. lowest_priority
..
.. PRIORITY'FIRST AND PRIORITY'LAST ARE RESERVED FOR USE BY THE KERNEL
.. ANY USE OF THESE VALUES WILL GENERATE INCORRECT PRIORITY ASSIGNMENTS
.. FROM THE APPLICATION VIEWPOINT
..
.. this abstraction can be built by the user on top of the Kernel

```



```

-- blocked:  unable to run as it is waiting for an event to occur
-- dead:      unable to run again, terminated
-- REQ: 9.1.23
--
type Process_State is (
    Running, Suspended, Blocked, Dead);
--
-- the default process state (i.e. upon process creation) is suspended
--
Default_Process_State : constant Process_State := Suspended;
end Generic_Schedule_Types;
pragma page;

```

```

-- primitives to gain greater protection
-- REQ: 5.1.6, 9.1.3, 9.1.4, 9.1.5
--
type Priority is new Hw_Natural range
0 .. Hw_Natural (Lowest_Priority_Value + 1);
--
-- the lowest priority possible
-- REQ: 5.1.6, 9.1.3
--
Highest_Priority : constant Priority := Priority_First + 1;
Lowest_Priority : constant Priority := Priority (Lowest_Priority_Value);
--
-- when a default priority is given on kernel primitives that could block,
-- this value is used to indicate that the process's current priority should
-- be assumed (the "ouge")
--
Current_Process_Priority : constant Priority := 0;
--
-- the default priority is always the lowest possible value
-- REQ: 5.1.6, 9.1.2
--
Default_Priority : constant Priority := Lowest_Priority;
--
-- this flag indicates whether or not a process may be preempted by another
-- process at the same priority level
-- REQ: 9.1.10
--
type Preemption is (Enabled, Disabled);
--
-- the default preemption is always that a process may be preempted
-- REQ: 9.1.2
--
Default_Preemption : constant Preemption := Enabled;
--
-- a process is always in one of 4 possible states
-- running: the process currently executing on the processor
-- suspended: eligible to be running but, due to its priority, not
--            currently running

```

```

.....
./ MODULE NAME:
./  schedule_types
./
./ MODULE TYPE:
./  package specification
./
./ MODULE PURPOSE:
./  an example of (and a template for) an instantiation of the generic
./  package Generic_schedule_types
./
./ MODULE DESCRIPTION:
./  package instantiating the schedule abstract types where:
./  error checking for a null priority range is enabled
./  highest priority value is (arbitrarily) defined as 1
./  lowest priority value is (arbitrarily) defined as 255
./
./ MODULE CONTENTS:
./  schedule_types (package specification)
./  current_process_priority (constant)
./  default_preemption (constant)
./  default_priority (constant)
./  default_process_state (constant)
./  highest_priority (constant)
./  lowest_priority (constant)
./  null_priority_range (exception)
./  preemption (type)
./  priority (type)
./  process_state (type)
./  schedule_types (package body)
./  elaboration-time initialization code
./
./ REFERENCES:
./  DESIGN DOCUMENTS:
./    Kernel Facilities Definition
./
./  USER'S MANUAL:
./    Kernel User's Manual
./
./ TESTING AND VALIDATION:
./  none
./
./ NOTES:
./  none
./
./ MODIFICATION HISTORY:
./  18aug88 bamberg created (complex clean)
./  13dec88 dvk deleted maximum slices per quantum and
./             ticks per slice. This functionality has been
./             moved to generic timeslice management.
./  15feb89 dvk updated documentation to reflect prior changes,
./             added highest priority to documentation . per
./             bug report 0008 SEI

```

```

./ 26apr89 dts changed lowest priority value to 255
./ for performance testing
./ 27aug89 bamberg final doc. check for 01sep89 freeze
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./
./ The following copyright must be included in this software and
./ all software utilizing this software
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....

```

with Generic Schedule Types:
 pragma Elaborate (Generic_Schedule_Types);
 .. to get: see MODULE CONTENTS

.....
 .. package short name: ST

package Schedule Types

```

is new Generic_Schedule_Types (
  Null_Priority_Range_Enabled => True,
  Lowest_Priority_Value => 255);
pragma (page);

```

```

--/ .....
--/ MODULE NAME:
--/   generic_network_globals
--/
--/ MODULE TYPE:
--/   generic package specification
--/
--/ MODULE PURPOSE:
--/   exports network global information
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   collection of globals related to network configuration but which can
--/   be used generally by anyone
--/
--/ MODULE CONTENTS:
--/   generic_network_globals (generic package specification)
--/   bus_address (type)
--/   bus_address_check (exception)
--/   bus_address_check_enabled (generic formal parameter)
--/   first_bus_address (constant)
--/   last_bus_address_value (pseudo generic formal parameter)
--/   last_bus_address (constant)
--/   last_bus_address_value (pseudo generic formal parameter)
--/   null_address (constant)
--/   null_address_value (generic formal parameter)
--/   process_index_type (type)
--/   processor_idenfier (type)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     Kernel Facilities Definition
--/     --REQ: 5.1.3; 5.1.4; 5.1.8; 6.1.11
--/     --REQ: 10.1.6; 10.1.15
--/     --PRIM: 15.1.4
--/     --TIME: N/A
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Integration testing with generic_processor_management
--/
--/ NOTES:
--/   none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (complies clean)
--/   29aug88 bamberg move bus_address stuff from GNC to break
--/               circularity (DGG, GPTB)
--/               add type: process_index_type
--/               hard-wire low bus_address and high bus_address
--/               to provide static bounds to type bus_address
--/               (as later used with rsp spec)

```

```

--/ 31aug88 dvk      changed process number in process_index_type to
--/                  be hw Integer (not hw_long Integer)
--/ 05oct88 bamberg  add initialization to process_index_type
--/ 07oct88 bamberg  reorder "bus_address" declarations to make types
--/                  types of low/high bus_address correct; added
--/                  two "bus_address" pseudo-generic to "nals" to
--/                  maintain consistency with the rest of Kernel
--/ 27aug89 bamberg  final doc. check for 01sep89 freeze
--/
--/-----
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/-----
pragma (page);

```

```

with Hardware_Interface; use Hardware_Interface;
-- to get: hw_natural

with Kernel_Exceptions;
-- to get: bus_address_check

generic
--
-- ERROR checking conditional compilation flags
--
-- REQ: 5.1.8
Bus_Address_Check_Enabled : In Boolean := True;

-- value indicating a null bus address (this value is never used within the
-- Kernel except for the initialization of the exported constant, nor should
-- it be used without the Kernel)
--
Null_Bus_Address_Value : In Hw_Natural;

Pragma(Page);
--/ package short name: GNG
--/-----
package Generic_Network_Globals
is
--
-- bus_address check raised by:
-- Initialization code in corresponding package body
--
--REQ: 5.1.3; 5.1.4; 6.1.11
Bus_Address_Check : exception
renames Kernel_Exceptions.Bus_Address_Check;
--
-- begin pseudo generic formal parameters ...
--
-- the following two values are hard-wired because ada does not allow
-- for non statically bounded types to be used within rep specs; the
-- type bus_address is used within a rep spec in datagram_globals; these
-- should be treated as generic formal parameters (i.e., they should NOT
-- be used outside this package)
--
-- value indicating the lowest possible bus address

```

```

First_Bus_Address_Value : constant Hw_Natural := 0;
-- value indicating the highest possible bus address
Last_Bus_Address_Value : constant Hw_Natural := 255;
-- *** end pseudo generic formal parameters ***
-- bus_address is a hw_integer with values within a user-specified range
--
-- PRIM: 15.1.4
type Bus_Address is new Hw_Natural
range First_Bus_Address_Value .. Last_Bus_Address_Value;
-- the following two declarations make available to the application the
-- first and the last bus_address values
First_Bus_Address : constant Bus_Address :=
Bus_Address(First_Bus_Address_Value);
Last_Bus_Address : constant Bus_Address :=
Bus_Address(Last_Bus_Address_Value);
-- this value is the null value for type bus_address
Null_Address : constant Bus_Address :=
Bus_Address(Null_Bus_Address_Value);
pragma (page);

-- a processor identifier (device identifier) is an index into the
-- NCT, and, as such, is a natural number
-- REQ: 10.1.6; 10.1.15
type Processor_Identifier is new Hw_Natural;
-- the process_index_type is another way to uniquely reference a process
-- components include:
-- node_number
-- the bus_address to which the processor corresponds
-- default value:
-- none
-- process_number
-- the number with respect to the node_number that uniquely identifies
-- this process
-- default value:
-- none
type Process_Index_Type is record
Node_Number : Bus_Address := Bus_Address'First;
Process_Number : Hw_Integer := Hw_Integer'First;
end record;
end Generic_Network_Globals;
pragma (page);

```

```

.....
--/ MODULE NAME:
--/   network_globals
--/
--/ MODULE TYPE:
--/   package specification
--/
--/ MODULE PURPOSE:
--/   an example of (and a template for) an instantiation of the generic
--/   package generic_network_globals
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   package instantiating the network_globals information where:
--/   checking for bus_address_check is enabled
--/   the value of null_bus_address_value is initialized to 16#00#
--/
--/ MODULE CONTENTS:
--/   network_globals (package specification)
--/   bus_address (type)
--/   bus_address_check (exception)
--/   first_bus_address (constant)
--/   last_bus_address (constant)
--/   null_address (constant)
--/   process_index_type (type)
--/   processor_identifier (type)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     Kernel Facilities Definition
--/     --REQ: 5.1.4, 5.1.7, 6.1.1
--/     --PRIM: N/A
--/     --TIME: N/A
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Integration testing with generic_processor_management
--/
--/ NOTES:
--/   none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (complex clean)
--/   25aug88 bamberg final doc. check for 01sep89 freeze
--/
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/   The following copyright must be included in this software and
--/   all software utilizing this software.

```

```

--/ .....
--/ DISTRIBUTION:
--/
--/   Approved for public release; distribution is unlimited.
--/
--/   Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/   PA. The Software Engineering Institute (SEI) is a federally
--/   funded research and development center established and operated
--/   by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/   Department of Defense under contract F19628-85-C-0003, the SEI
--/   is supported by the services and defense agencies, with the U.S.
--/   Air Force as the executive contracting agent.
--/
--/   Permission to use, copy, modify, or distribute this software and
--/   its documentation for any purpose and without fee is hereby
--/   granted, provided that the above copyright notice appear in all
--/   copies and that both that copyright notice and this permission
--/   notice appear in supporting documentation. Further, the names
--/   of Software Engineering Institute or Carnegie Mellon University may
--/   not be used in advertising or publicity pertaining to
--/   distribution of the software without specific, written prior
--/   permission. CMU makes no claims or representations about the
--/   suitability of this software for any purpose. This software is
--/   provided "as is" and no warranty, express or implied, is made
--/   by the SEI or CMU, as to the accuracy and functioning of the
--/   program and related program material, nor shall the fact of
--/   distribution constitute any such warranty. No responsibility is
--/   assumed by the SEI or CMU in connection herewith.
--/   .....

```

prgma (page).

```

with Generic_Network_Globals;
-- to get see MODULE CONTENTS

--/ .....
--/ package short name: NG
--/ .....

package Network_Globals

is new Generic_Network_Globals (
    Bus_Address_Check_Enabled => True,
    Null_Bus_Address_Value => 16#00#);
pragma (page);

```

```

--/ .....
--/ MODULE NAME:
--/ generic_network_configuration
--/
--/ MODULE TYPE:
--/ generic_package_specification
--/
--/ MODULE PURPOSE:
--/ network_configuration_table (NCT) abstraction
--/
--/ .....
--/ MODULE DESCRIPTION:
--/ provides the abstraction of the NCT and its component parts
--/
--/ .....
--/ MODULE CONTENTS:
--/ generic_network_configuration (generic_package_specification)
--/ configuration_table (type)
--/ get_processor_id (function)
--/ maximum_length_of_processor_name (constant)
--/ maximum_length_of_processor_name_value (generic_formal_parameter)
--/ NCT (data structure)
--/ NCT entry (type)
--/ number_of_nodes_value (generic_formal_parameter)
--/ number_of_nodes (constant)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ Kernel_Facilities_Definition
--/   --REQ: 5.1.6; 6.1.6; 6.1.10; 6.1.11;
--/   --REQ: 7.1.14; 7.1.16; 10.1.6; 10.1.15; 10.1.26
--/   --PRIM: 15.1.4
--/   --TIME: 6.2.1
--/
--/ USER'S MANUAL:
--/ Kernel_User's_Manual
--/
--/ TESTING AND VALIDATION:
--/ Integration_testing_with_generic_processor_management
--/
--/ NOTES:
--/ none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (complies clean)
--/ 24aug88 bamberg change type of Index of NCT to
--/   network_globals_processor_identifier (which it
--/   should have been) and begin NCT origin at 1
--/ 29aug88 bamberg move bus address stuff to GNG to break
--/   circularity (DOG, GPTB)
--/ 01sep88 bamberg export maximum_length_of_processor_name
--/ 13dec88 rvs provide complete default initialization for
--/   NCT entries
--/ 22mar89 rvs move get_processor_id from PIT to here
--/ 30may89 rvs Bug fix 0063 sei

```

```

../ 27aug89 bamberg final.doc: check for 01sep89 freeze
../
../-----
../ VERSION: 3.0
../
../DISCLAIMER:
../
../The following copyright must be included in this software and
../all software utilizing this software.
../
../DISTRIBUTION:
../
../Approved for public release, distribution is unlimited.
../
../Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../PA. The Software Engineering Institute (SEI) is a federally
../funded research and development center established and operated
../by Carnegie Mellon University (CMU). Sponsored by the U.S.
../Department of Defense under contract F19628-85-C-0003, the SEI
../is supported by the services and defense agencies with the U.S.
../Air Force as the executive contracting agent.
../
../Permission to use, copy, modify, or distribute this software and
../its documentation for any purpose and without fee is hereby
../granted, provided that the above copyright notice appear in all
../copies and that both that copyright notice and this permission
../notice appear in supporting documentation. Further, the names
../Software Engineering Institute or Carnegie Mellon University may
../not be used in advertising or publicity pertaining to
../distribution of the software without specific, written prior
../permission. CMU makes no claims or representations about the
../suitability of this software for any purpose. This software is
../provided "as is" and no warranty, express or implied, is made
../by the SEI or CMU, as to the accuracy and functioning of the
../program and related program material, nor shall the fact of
../distribution constitute any such warranty. No responsibility is
../assumed by the SEI or CMU in connection herewith.
../

```

pragma (page);

```

with Hardware Interface; use Hardware_Interface;
-- to get: hw_natural
--
-- hw_positive
-- hw_string

with Network_Globals;
-- to get: bus_address
-- null_address
-- processor_Identifier

with Process_Table;
-- to get: null_process
-- process_Identifier

generic
--
-- the maximum number of nodes that this hardware configuration can handle
-- (this value defines the numbers of entries in the NCT)
-- REQ: 5.1.6; 6.1.10; 6.1.11; 7.1.14; 7.1.16; 10.1.6; 10.1.15; 10.1.26
-- PRIM: 15,1,4
-- TIME: 6.2.1
--
Number_Of_Nodes_Value : In Hw_Positive;
--
-- the maximum length of the string used to represent a logical processor
-- name (i.e., upper bound on a string)
--
-- note that null strings are allowed, so the application need not carry
-- extra string space around with it
--
Maximum_Length_Of_Processor_Name_Value : In Hw_Natural;

Pragma(Page);
../-----
../package short name: GNC
../-----

package Generic_Network_Configuration
is
--
-- the maximum number of nodes that this hardware configuration can handle
-- (this value defines the numbers of entries in the NCT)
--
Number_Of_Nodes : constant Hw_Positive := Number_Of_Nodes_Value;
--
-- the maximum length of the string used to represent a logical processor

```



```

-- name (i.e., upper bound on a string)
--
-- Maximum Length Of Processor Name : constant Hw_Natural :=
-- Maximum Length Of Processor Name_Value;
--
-- each entry in the network control table (NCT) describes the physical
-- connectivity of one node in the network
--
-- the information needed about each node is:
-- logical name
-- the string-valued name given by the application engineer (this is
-- mapped to the device_ID, which is just an index into the NCT -
-- during initialization, once that is done, the Kernel refers to
-- the processor by device_ID)
--
-- default value:
-- none
--
-- this value should never change after initialization in Main Unit
--
-- physical address
-- the actual bus address at which the device is located; this value
-- is used in the datagram packet wrapper to identify the network
-- node that is to receive the packet containing the datagram
--
-- default value:
-- null_address
--
-- this value should never change after initialization in Main Unit
--
-- Kernel device
-- indication of whether or not the device is also running the Kernel
-- (e.g., whether or not the device responds to the Kernel
-- communication protocols)
--
-- possible values:
-- true (the processor is running the Kernel)
-- false (the processor is not running the Kernel)
--
-- default value
-- true (the processor is running the Kernel)
--
-- this value should never change after initialization in Main Unit
--
-- needed to run
-- indication of whether or not the device must successfully complete
-- the Kernel initialization protocol (i.e., always should be false
-- when Kernel device is false)
--
-- possible values:
-- true (the device does participate in the initialization
-- protocol)
-- false (the device does not participate in the initialization

```

```

-- protocol)
--
-- default value:
-- true (the device does participate in the initialization
-- protocol)
--
-- this value should never change after initialization in Main Unit
--
-- allocated process ID
-- identifier of "surrogate process" allocated to receive messages
-- from the specified non-Kernel device.
--
-- default value:
-- null
--
-- this value may change via a call to the Kernel primitives:
-- allocate_device_receiver
-- die
-- kill
--
-- initialization_order
-- order in which the processors identified in the NCT are to be
-- initialized
--
-- default value:
-- 0 (all NCT entries have the same initial value; initialization
-- proceeds following each entry in the NCT)
--
-- this value should never change after initialization in Main Unit
--
-- initialization_complete
-- indication of whether or not the initialization protocol for this
-- processor has been completed
--
-- possible values:
-- true (initialization has completed)
-- false (initialization has not completed)
--
-- default value:
-- false (initialization has not completed)
--
-- this value is set by the Kernel during initialization and should
-- never change after initialization is complete
--
--
-- type Nct Entry is record
-- Logical Name : Hw_String
-- (1 .. Positive (Maximum Length Of Processor Name Value))
-- := (others => '');
-- Physical Address : Network_Globals.Bus_Address :=
-- Network_Globals.Null_Address;
-- Kernel Device : Boolean := True;
-- Needed To Run : Boolean := True;
-- Allocated Process ID : Process_Table.Process_Identifier := null;

```

```

Initialization_Order : Hw Natural := 0;
Initialization_Complete : Boolean := False;
end record;

-- template for the NCT: an array (bounds user-defined) of NCT_entry
-- defined above
--
type Configuration_Table is array (
  Network_Globals.Processor_Identifier range <>) of Nct_Entry;

-- this declares the actual NCT with the user-defined number of nodes
-- in it; the user assigns values to each entry in the NCT at compile time
--
-- *****
-- once this table is declared, the application should NOT make ANY changes
-- to it; EVER; proper Kernel functioning is NOT guaranteed if any changes
-- are made outside of calls to Kernel primitives
-- *****
--
Nct : Configuration_Table (Network_Globals.Processor_Identifier
  range 1 .. Network_Globals.Processor_Identifier (Number_Of_Nodes));

pragma (page);

```

```

function Get_Processor_Id (Node_Address: in Network_Globals.Bus_Address)
return Network_Globals.Processor_Identifier;
pragma inline (Get_Processor_Id);
-- *****
-- SUBPROGRAM NAME:
-- get_processor_id
--
-- DESCRIPTION:
-- Locate a node's entry in the NCT based on it's network address.
--
-- NOTES:
-- The need for this function arises because the entry for a node in
-- the NCT is not required to correspond to its network address.
-- Thus, a node with address 0 may be located in entry 9 of the NCT.
--
-- Also, care must be exercised when using this function. The NCT must
-- be initialized for this routine to return the correct result. Since
-- that can not be guaranteed until after a call to one of the
-- initialize entries in processor_management, this function should
-- not be used during package elaboration.
--
-- REQ: 6.1.11, 10.1.6, 10.1.15
--
-- PRIM: N/A
--
-- TIME: 6.2.1
--
PARAMETERS:
node_address
the network address of the node
default value:
none
return
the location of the node's entry in the NCT
default value:
none
PRECONDITIONS:
A valid node address is supplied
ACTIONS:
Search the NCT for the node address
Return the NCT location of the node
POSTCONDITIONS:
The returned location is a valid NCT entry
ERROR PROCESSING:
None
-- *****

```

end Generic_Network_Configuration;
pragma (page);

```

--/ .....
--/ MODULE NAME:
--/   network_configuration
--/
--/ MODULE TYPE:
--/   package specification
--/
--/ MODULE PURPOSE:
--/   network configuration table (NCT) abstraction
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   provides the abstraction of the NCT and its component parts
--/
--/ MODULE CONTENTS:
--/   network_configuration (package specification)
--/   configuration_table (type)
--/   get_processor_id (function)
--/   maximum_length_of_processor_name (constant)
--/   maximum_length_of_processor_name_value
--/   (pseudo-generic formal parameter)
--/   NCT (data structure)
--/   NCT_entry (type)
--/   number_of_nodes_value (pseudo-generic formal parameter)
--/   number_of_nodes (constant)
--/
--/ REFERENCES:
--/   Kernel Facilities Definition
--/   --REQ: 5.1.6; 6.1.10; 6.1.11; 7.1.14; 7.1.16;
--/   --REQ: 10.1.6; 10.1.15; 10.1.26
--/   --PRM: 15.1.4
--/   --TIME: 6.2.1
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Integration testing with generic_processor_management
--/
--/ NOTES:
--/   none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (compiles clean)
--/   24aug88 bamberg change type of index of NCT to
--/   network_globals_processor_identities (which it
--/   should have been) and begin NCT origin at 1
--/   29aug88 bamberg move bus address stuff to GNG to break
--/   circularity (DGG, GPTB)
--/   01sep88 bamberg HAND INSTANTIATION due to telesoft bug!!!
--/   export maximum_length_of_processor_name
--/   13dec88 rlv provide complete default initialization for
--/   NCT entries
--/   22mar89 rlv move get_processor_id from PIT to here

```

```

../ 30may89 rns      bug fix 0063 sei
../ 25aug89 bamberg final doc. check for 01sep89 heeze
../ .....
../ VERSION: 3.0
../
../ DISCLAIMER:
../
../ The following copyright must be included in this software and
../ all software utilizing this software.
../
../ DISTRIBUTION:
../
../ Approved for public release, distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../ .....

```

pragma (page);

```

with Hardware_Interface; use Hardware_Interface;
-- to get: hw_natural
--      hw_positive
--      hw_string

with Network_Globals;
-- to get: bus_address
--      null_address
--      processor_identifier

with Process_Table;
-- to get: null_process
--      process_identifier

--/ .....
--/ package short name: NC
--/ .....
package Network_Configuration
is
--/
--/ *** BEGIN PSEUDO-GENERIC PARAMETERS ***
--/
--/ the maximum number of nodes that this hardware configuration can handle
--/ (this value defines the numbers of entries in the NCT)
--REQ: 5.1.6; 6.1.10; 8.1.11; 7.1.14; 7.1.16; 10.1.6; 10.1.15; 10.1.26
--PRIM: 15.1.4
--TIME: 6.2.1
--/
Number_Of_Nodes_Value : constant Hw_Positive := 4;

-- the maximum length of the string used to represent a logical processor
-- name (i.e., upper bound on a string)
-- note that null strings are allowed, so the application need not carry
-- extra string space around with it

Maximum_Length_Of_Processor_Name_Value : constant Hw_Natural := 16;

--/ *** END PSEUDO-GENERIC PARAMETERS ***
--/
-- the maximum number of nodes that this hardware configuration can handle
-- (this value defines the numbers of entries in the NCT)

```

Number Of Nodes : constant Hw_Positive := Number Of Nodes Value;

.. the maximum length of the string used to represent a logical processor
.. name (i.e., upper bound on a string)

Maximum_Length_Of_Processor_Name : constant Hw_Natural :=
Maximum_Length_Of_Processor_Name_Value;

.. each entry in the network control table (NCT) describes the physical
.. connectivity of one node in the network

.. the information needed about each node is:

.. logical name
.. the string-valued name given by the application engineer (this is
.. mapped to the device ID - which is just an index into the NCT -
.. during initialization, once that is done, the Kernel refers to
.. the processor by device ID)

.. default value:
.. none

.. this value should never change after initialization in Main Unit

.. physical address

.. the actual bus address at which the device is located; this value
.. is used in the datagram packet wrapper to identify the network
.. node that is to receive the packet containing the datagram

.. default value:
.. null address

.. this value should never change after initialization in Main Unit

.. Kernel device

.. indication of whether or not the device is also running the Kernel
.. (e.g., whether or not the device responds to the Kernel
.. communication protocols)

.. possible values:

.. true (the processor is running the Kernel)
.. false (the processor is not running the Kernel)

.. default value:

.. true (the processor is running the Kernel)

.. this value should never change after initialization in Main Unit

.. needed to run

.. indication of whether or not the device must successfully complete
.. the Kernel initialization protocol (i.e., always should be false
.. when Kernel device is false)

.. possible values:
.. true (the device does participate in the initialization
.. protocol)
.. false (the device does not participate in the initialization
.. protocol)

.. default value:
.. true (the device does participate in the initialization
.. protocol)

.. this value should never change after initialization in Main Unit

.. allocated process ID
.. identifier of "surrogate process" allocated to receive messages
.. from the specified non-Kernel device

.. default value:
.. null

.. this value may change via a call to the Kernel primitives:
.. allocate device receiver
.. die
.. kill

.. initialization order
.. order in which the processors identified in the NCT are to be
.. initialized

.. default value:
.. 0 (all NCT entries have the same initial value; initialization
.. proceeds following each entry in the NCT)

.. this value should never change after initialization in Main Unit

.. initialization complete
.. indication of whether or not the initialization protocol for this
.. processor has been completed

.. possible values:
.. true (initialization has completed)
.. false (initialization has not completed)

.. default value:
.. false (initialization has not completed)

.. this value is set by the Kernel during initialization and should
.. never change after initialization is complete

.. REQ 6 1 10, 10 1 6, 10 1 26
.. PRIM 15 1 4

.. type Nct Entry is record
.. Logical Name : Hw_String
.. (1..Positive (Maximum_Length_Of_Processor_Name_Value))

```

:= (others => ' ');
Physical_Address : Network_Globals.Bus_Address :=
Network_Globals.Null_Address;
Kernel_Device : Boolean := True;
Needed_To_Run : Boolean := True;
Allocated_Process_Id : Process_Table.Process_Identifier := null;
Initialization_Order : Hw_Natural := 0;
Initialization_Complete : Boolean := False;
end record;

-- template for the NCT: an array (bounds user-defined) of NCT_entry
-- defined above
--REQ 6.1.10, 10.1.6, 10.1.26
--PRIM: 15.1.4

type Configuration_Table is array (
Network_Globals.Process_Identifier range <>) of Nct_Entry;

-- this declares the actual NCT with the user-defined number of nodes
-- in it, the user assigns values to each entry in the NCT at compile time
-- .....
-- once this table is declared, the application should NOT make ANY changes
-- to it, EVER; proper Kernel functioning is NOT guaranteed if any changes
-- are made outside of calls to Kernel primitives
-- .....
--REQ 6.1.10, 10.1.6, 10.1.26
--PRIM: 15.1.4

Nct : Configuration_Table (Network_Globals.Process_Identifier
range 1 .. Network_Globals.Process_Identifier (Number_Of_Nodes));
pragma (page);

```

```

function Get_Processor_Id (Node_Address: in Network_Globals.Bus_Address)
return Network_Globals.Process_Identifier;
pragma Inline (Get_Processor_Id);
-- .....
--/ SUBPROGRAM NAME:
--/ get_processor_id
--/
--/ DESCRIPTION:
--/ Locate a node's entry in the NCT based on it's network address.
--/
--/ NOTES:
--/ The need for this function arises because the entry for a node in
--/ the NCT is not required to correspond to its network address.
--/ Thus, a node with address 0 may be located in entry 9 of the NCT.
--/
--/ Also, care must be exercised when using this function. The NCT must
--/ be initialized for this routine to return the correct result. Since
--/ that can not be guaranteed until after a call to one of the
--/ initialize entries in processor management, this function should
--/ not be used during package elaboration.
--/
--/ --REQ: 6.1.11, 10.1.6, 10.1.15
--/ --PRIM: N/A
--/
--/ --TIME: 6.2.1
--/
--/ PARAMETERS:
--/ node_address
--/ the network address of the node
--/
--/ default_value:
--/ none
--/
--/ return
--/ the location of the node's entry in the NCT
--/
--/ default_value:
--/ none
--/
--/ PRECONDITIONS:
--/ A valid node address is supplied
--/
--/ ACTIONS:
--/ Search the NCT for the node address
--/ Return the NCT location of the node
--/
--/ POSTCONDITIONS:
--/ The returned location is a valid NCT entry
--/
--/ ERROR PROCESSING:
--/ None
--/
--/

```

end Network Configuration;
pragma (page);

```

--/ .....
--/ MODULE NAME:
--/   generic processor management
--/
--/ MODULE TYPE:
--/   generic package specification
--/
--/ MODULE PURPOSE:
--/   perform the initialization of the physical network.
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   collection of subprograms initialize the physical network.
--/
--/ .....
--/ MODULE CONTENTS:
--/   generic processor management (generic package specification)
--/   calling unit not Main_Unit (exception)
--/   calling unit not Main_Unit_enabled (generic formal parameter)
--/   configuration tables inconsistent (exception)
--/   final sync initialization timeout expired (exception)
--/   initialization complete (procedure)
--/   initialize Master processor (procedure)
--/   initialize subordinate processor (procedure)
--/   Master initialization timeout expired (exception)
--/   network failure (exception)
--/   process initialization failure (exception)
--/   process maximum exceeded (exception)
--/   processor failed to ACK go message (exception)
--/   processor failed to transmit NCT (exception)
--/   subordinate initialization timeout expired (exception)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     Kernel Facilities Definition
--/       --REQ: 5.1.8; 6.1.8; 6.1.7; 6.1.8; 6.1.11
--/       --PRIM: 15.1.1; 15.1.2; 16.1.3
--/       --PRIM: N/A
--/       --TIME: N/A
--/
--/   USER'S MANUAL:
--/     Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Unit testing
--/
--/ NOTES:
--/   none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (compiles clean)
--/   29aug88 bamberg move maximum number of processes value into GPTB
--/   18nov88 rvs implemented all functionality
--/

```

```

--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
pragma (page);

```

```

with Kernel_Exceptions;
-- to get: calling unit not Main_Unit
-- configuration tables inconsistent
-- final sync initialization timeout expired
-- Master initialization timeout expired
-- network failure
-- process initialization failure
-- process maximum exceeded
-- processor failed to ACK go message
-- processor failed to transmit NCT
-- subordinate initialization timeout expired

with Hardware_Interface; use Hardware_Interface;
-- to get: hw duration
-- hw positive

with Time_Globals;

generic

-- ERROR checking conditional compilation flags
--
-- REQ: 5.1.8
--
Calling_Unit_Not_Main_Unit_Enabled : In Boolean := True;

Pragma(Page);
-- .....
-- package short name: GRM
-- .....

package Generic_Processor_Management
is
package Ke
package Kg
renames Kernel_Exceptions;
package Tg
renames Time_Globals;

-- REQ: 6.1.6; 6.1.7; 6.1.8; 6.1.11
--
-- calling unit not Main_Unit raised by:
-- Initialize Master processor
-- Initialize subordinate processor
-- Initialization complete
-- PRIM: 15.1.1; 15.1.2; 16.1.3

Calling_Unit_Not_Main_Unit : exception

```



```

renames Ke.Calling_Unit_Not_Main_Unit;
..
..
.. configuration_tables_inconsistent_raised_by:
.. .. initialize_master_processor
.. .. PRIM: 15.1.1
Configuration_Tables_Inconsistent : exception
renames Ke.Configuration_Tables_Inconsistent;
..
.. final_sync_initialization_timeout_expired_raised_by:
.. .. initialization_complete
.. .. PRIM: 16.1.3
Final_Sync_Initialization_Timeout_Expired : exception
renames Ke.Final_Sync_Initialization_Timeout_Expired;
..
.. master_initialization_timeout_expired_raised_by:
.. .. initialize_master_processor
.. .. PRIM: 15.1.1
Master_Initialization_Timeout_Expired : exception
renames Ke.Master_Initialization_Timeout_Expired;
..
.. network_failure_raised_by:
.. .. initialize_master_processor
.. .. initialize_subordinate_processor
.. .. initialization_complete
.. .. PRIM: 15.1.1
.. .. PRIM: 15.1.2
.. .. PRIM: 16.1.3
Network_Failure : exception
renames Ke.Network_Failure;
..
.. process_initialization_failure_raised_by:
.. .. initialization_complete
.. .. PRIM: 16.1.3
Process_Initialization_Failure : exception
renames Ke.Process_Initialization_Failure;
..
.. process_maximum_exceeded_raised_by:
.. .. initialization_complete

```

```

.. .. PRIM: 16.1.3
Process_Maximum_Exceeded : exception
renames Ke.Process_Maximum_Exceeded;
..
.. processor_failed_to_ack_go_message_raised_by:
.. .. initialize_master_processor
.. .. PRIM: 15.1.1
Processor_Failed_To_Ack_Go_Message : exception
renames Ke.Processor_Failed_To_Ack_Go_Message;
..
.. processor_failed_to_transmit_NCT_raised_by:
.. .. initialize_master_processor
.. .. PRIM: 15.1.1
Processor_Failed_To_Transmit_Nct : exception
renames Ke.Processor_Failed_To_Transmit_Nct;
..
.. subordinate_initialization_timeout_expired_raised_by:
.. .. initialize_subordinate_processor
.. .. PRIM: 15.1.2
Subordinate_Initialization_Timeout_Expired : exception
renames Ke.Subordinate_Initialization_Timeout_Expired;
pragma (page);

```

```

procedure Initialize_Master_Processor (Base_Epoch: In TgEpoch_Time;
                                         Timeout_After : In TgElapsed_Time);
--=====
-- SUBPROGRAM NAME:
-- Initialize_Master_processor
--
-- DESCRIPTION:
-- begins Master Initialization sequence, identifies caller as Master
-- processor for duration of Initialization only
--
-- --REQ: 6.1.1; 6.1.3; 6.1.4; 6.1.5
--
-- --PRIM: 15.1.1
--
-- --TIME: 6.2.1; 6.2.2
--
-- PARAMETERS:
-- base epoch
-- the base epoch time for the network
--
-- default value:
-- none
--
-- timeout
-- duration to wait for a subordinate to respond to a message
-- from the Master processor.
--
-- default value:
-- none
--
-- NOTES:
-- Only those nodes that are Kernel devices and needed to run can
-- result in a network failure.
--
-- PRECONDITIONS:
-- the network hardware is up and initialized
-- only the processor intending to be Master for initialization purposes
-- may call this procedure
-- no other processor has yet called Initialize_Master_processor
-- a previous call of Initialize_* processor by this processor must not
-- have been done
--
-- ACTIONS:
-- Initialize all internal Kernel data structures
-- Master identifies itself to all subordinate processors
-- Master waits for all subordinates to send back their NCTs
-- Master compares other NCTs with its own to ensure a network-wide
-- consistent view of the physical network topology
-- Master tells all subordinate processors the current time of day
-- Master waits for all subordinates to acknowledge the go message
--
-- POSTCONDITIONS:
-- all NCTs are consistent (i.e., the physical topology is consistent
-- across processors)
-- all the clocks are synchronized

```

```

--/ the Master/subordinate Initialization has been dissolved
--/
--/ ERROR PROCESSING:
--/ when the call is detected from a unit that is not the Main Unit:
--/ the Kernel raises calling_unit_not_Main_Unit exception
--/ POSTCONDITIONS:
--/ the requested action does not take place
--/ Master informs all processors that there is a network failure
--/ the Kernel raises configuration_tables_inconsistent exception
--/ and control is returned to the Main Unit to inform the external
--/ world that initialization has failed
--/ POSTCONDITIONS:
--/ the network is not functional
--/ when the Master's initialization parameter has expired:
--/ Master informs all processors that there is a network failure
--/ the Kernel raises Master_initialization_timeout_expired exception
--/ and control is returned to the Main Unit to inform the external
--/ world that initialization has failed
--/ POSTCONDITIONS:
--/ the network is not functional
--/ when a processor fails to acknowledge the Master's "go" message:
--/ Master informs all processors that there is a network failure
--/ the Kernel raises processor_failed_to_ACK_go_message exception
--/ and control is returned to the Main Unit to inform the external
--/ world that initialization has failed
--/ POSTCONDITIONS:
--/ the network is not functional
--/ when an NCT is not received from a processor that is required to
--/ participate in the initialization protocol:
--/ Master informs all processors that there is a network failure
--/ the Kernel raises processor_failed_to_transmit_NCT exception
--/ and control is returned to the Main Unit to inform the external
--/ world that initialization has failed
--/ POSTCONDITIONS:
--/ the network is not functional
--/
--=====
pragma (page);
--=====

```

```

procedure Initialize Subordinate Processor (Timeout Alter : in Tg Elapsed Time);
-- SUBPROGRAM NAME:
-- initialize_subordinate_processor
--
-- DESCRIPTION:
-- begins subordinate initialization sequence; identifies caller as a
-- subordinate processor for duration of initialization only
--
-- REQ: 6.1.2; 6.1.3; 6.1.4
--
-- PRIM: 15.1.2
--
-- TIME: 6.2.1
--
-- PARAMETERS:
-- timeout
-- duration to wait until subordinate asserts that network
-- initialization has failed
--
-- default value:
-- none
--
-- PRECONDITIONS:
-- the network hardware is up and initialized
-- only the processors not intending to be Master for initialization
-- purposes may call this procedure
-- a previous call of initialize_subprocessor by this processor must not
-- have been done
--
-- ACTIONS:
-- initialize all internal kernel data structures
-- a subordinate waits for the Master to identify itself
-- a subordinate sends its NCT to the Master for consistency checking
-- a subordinate receives the epoch time from the Master and sets its
-- own local time
--
-- POSTCONDITIONS:
-- all NCTs are consistent
-- The local clock is synchronized with the Master's clock
--
-- ERROR PROCESSING:
-- when the call is detected from a unit that is not the Main Unit:
-- the kernel raises calling_unit_not_Main_Unit_exception
-- POSTCONDITIONS:
-- the requested action does not take place
-- when the subordinate's initialization timeout has expired:
-- the subordinate processor informs all processors that there is a
-- network failure
-- the kernel raises
-- subordinate_initialization_timeout_expired_exception and
-- control is returned to the Main Unit to inform the external
-- world that initialization has failed
-- POSTCONDITIONS:
-- the network is not functional

```

```

-- when the subordinate receives an out of sequence message from
-- the Master
-- the subordinate processor informs all processors that there is a
-- network failure
-- the kernel raises the network_failure_exception and
-- control is returned to the Main Unit to inform the external
-- world that initialization has failed
-- POSTCONDITIONS:
-- the network is not functional

```

pragma (page);

```

procedure Initialization_Complete (Timeout_After : in TgElapsed_Time);
--=====
-- SUBPROGRAM NAME:
-- Initialization_Complete
--
-- DESCRIPTION:
-- asserts that the declaration and creation of all processes on this
-- processor is complete
-- --REQ: 7.1.15, 7.1.16, 7.1.17, 7.1.25, 7.1.26
-- --PRIM: 16.1.3
-- --TIME: 6.2.1
-- PARAMETERS:
-- timeout
-- duration to wait until this processor asserts that its
-- initialization has failed
-- default value:
-- none
--
-- PRECONDITIONS:
-- all declare_process and create_process calls have been completed for
-- this processor
--
-- ACTIONS:
-- the Kernel informs all other Kernels that it has completed
-- initialization
-- wait for all other nodes to complete initialization (i.e.,
-- invoke this primitive)
-- the Kernel checks internal data structures for consistency and
-- completeness
-- the Kernel de-schedules the Main Unit
-- the Kernel Scheduler is given control of the processor
--
-- POSTCONDITIONS:
-- processor-specific logical topology is consistent and complete
-- Kernel data structures entries are consistent
-- the Main Unit is no longer a schedulable entity
--
-- ERROR PROCESSING:
-- when the call is detected from a unit that is not the Main Unit:
-- the Kernel raises calling_unit_not_Main_Unit_exception
-- POSTCONDITIONS:
-- the requested action does not take place
-- when the Kernel detects that the number of communication partners
-- and the number of locally created processes exceeds the
-- user-specified maximum:
-- the Kernel informs all processors that there is a network failure
-- the Kernel raises process_maximum_exceeded_exception
-- POSTCONDITIONS:
-- the network is not functional
-- the Kernel raises remote_Kernel_process_undefined_exception

```

```

-- when the initialization complete parameter has expired:
-- the Kernel informs all processors that there is a network failure
-- (which could simply be that the other processors haven't
-- yet indicated that they have completed initializing)
-- the Kernel raises final_sync_initialization_timeout_exception
-- POSTCONDITIONS:
-- the network is not functional
-- when the Kernel detects that a single named process has been
-- created on more than one processor or
-- when the Kernel detects that an expected remote communication partner
-- was not created on the remote processor:
-- the Kernel informs all processors that there is a network failure
-- the Kernel raises process_initialization_failure_exception
-- POSTCONDITIONS:
-- the network is not functional
--
--=====
end Generic_Processor_Management;
pragma page;

```

```

.....
MODULE NAME
processor_management

MODULE TYPE
package_specification

MODULE PURPOSE
an example of (and a template for) an instantiation of the generic
package_generic_processor_management

MODULE DESCRIPTION
package_instantiating the creation and maintenance of the physical
network configuration capabilities where:
error checking for calling_unit_not_Main_Unit is enabled

MODULE CONTENTS
processor_management (package_specification)
calling_unit_not_Main_Unit (exception)
configuration_tables_inconsistent (exception)
final_sync_initialization_timeout_expired (exception)
initialization_complete (procedure)
initialize_master_processor (procedure)
initialize_subordinate_processor (procedure)
master_initialization_timeout_expired (exception)
network_failure (exception)
process_initialization_failure (exception)
process_madrum_exceeded (exception)
processor_failed_to_ack_go_message (exception)
processor_failed_to_transmit_NCT (exception)
start_subordinate_processors (procedure)
subordinate_initialization_timeout_expired (exception)

REFERENCES
DESIGN DOCUMENTS:
Kernel_Facilities_Definition

USER'S MANUAL
Kernel_User's_Manual

TESTING AND VALIDATION:
none

NOTES:
none

MODIFICATION HISTORY:
18aug88 bamberg created (compiles clean)
18nov88 rty deleted unneeded exceptions and generic
formal parameters

VERSION: 3.0

```

```

DISCLAIMER:
The following copyright must be included in this software and
all software utilizing this software.

DISTRIBUTION:
Approved for public release; distribution is unlimited.

Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
PA. The Software Engineering Institute (SEI) is a federally
funded research and development center established and operated
by Carnegie Mellon University (CMU). Sponsored by the U.S.
Department of Defense under contract F19628-85-C-0003, the SEI
is supported by the services and defense agencies, with the U.S.
Air Force as the executive contracting agent.

Permission to use, copy, modify, or distribute this software and
its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission
notice appear in supporting documentation. Further, the names
of Software Engineering Institute or Carnegie Mellon University may
not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission. CMU makes no claims or representations about the
suitability of this software for any purpose. This software is
provided "as is" and no warranty, express or implied, is made
by the SEI or CMU, as to the accuracy and functioning of the
program and related program material, nor shall the fact of
distribution constitute any such warranty. No responsibility is
assumed by the SEI or CMU in connection herewith.

with Generic Processor Management;
pragma Elaborate (Generic Processor Management);
.. to get: see MODULE CONTENTS

package_short_name: RM
package_Generic_Protessor_Management

is new Generic Processor Management (
Calling Unit Not Main Unit Enabled => False);
pragma (page);

```

```

.....
--/ MODULE NAME:
--/   generic_process_managers_globals
--/
--/ MODULE TYPE:
--/   generic_package_specification
--/
--/ MODULE PURPOSE:
--/   collection of types used by process_manager subprograms
--/
--/
--/ MODULE DESCRIPTION:
--/   collection of types used by process_manager subprograms
--/
--/ MODULE CONTENTS:
--/   generic_process_managers_globals (generic package specification)
--/   device_name_type (type)
--/   how_to_handle_message_queue_overflow (type)
--/   maximum_length_of_process_name (constant)
--/   maximum_length_of_process_name_value (generic formal parameter)
--/   process_name_type (type)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     Kernel Facilities Definition
--/       --REQ: 10.1.22
--/       --PRIM: N/A
--/       --TIME: N/A
--/   USER'S MANUAL:
--/     Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Integration testing with generic_process_managers
--/
--/ NOTES:
--/   none
--/
--/
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (compiles clean)
--/   23may89 iac   Bug 0048 SEI; add overwrite oldest message
--/              enumeration literal to 'overflow' type
--/   27aug89 bamberg final doc check for 01sep89 freeze
--/
--/
--/ VERSION: 3 0
--/
--/ DISCLAIMER:
--/   The following copyright must be included in this software and
--/   all software utilizing this software
--/
--/ DISTRIBUTION:
--/

```

```

--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....

```

pragma (page):

```

-- this type of string is used to indicate a process name
--
type Process_Name_Type is new Hw_String (
  1 .. Positive (Maximum_Length_Of_Process_Name) );
-- this type of string is used to indicate a non-Kernel device name
--
type Device_Name_Type is new Hw_String (
  1 .. Positive (Maximum_Length_Of_Process_Name) );
end Generic_Process_Managers_Globals;
pragma (page);

```

```

with Hardware_Interface; use Hardware_Interface;
-- to get: hw_natural
-- hw_string
generic
-- the maximum number of characters of a process name that is stored in the
-- process table (i.e., upper bound on a string)
-- note that null strings are allowed, so the application need not carry
-- extra string space around with it
Maximum_Length_Of_Process_Name_Value : in Hw_Natural;
Pragma(Page);
-- package short name: GPMG
package Generic_Process_Managers_Globals

```

```

is
-- indication of how to handle message queue overflow (should it occur)
-- choices are:
-- drop_newest_message
-- if the incoming message is larger than the available space in the
-- message queue, the incoming message is "dropped," and all required
-- kernel-to-Kernel negative acknowledgements are automatically made
-- overwrite_oldest_message
-- if the message queue is full when a message is received, the oldest
-- message is to be removed from the message queue and all required
-- kernel-to-Kernel negative acknowledgements are automatically made
-- for the removed message.
-- user-selectable option on a per-process basis at process creation time
-- REQ: 10.1.22
type How_To_Handle_Message_Queue_Overflow is (
  Drop_Newest_Message,
  Overwrite_Oldest_Message );
-- the maximum number of characters of a process name that is stored in the
-- process table (i.e., upper bound on a string); this value is exported
Maximum_Length_Of_Process_Name : constant Hw_Natural :=
Maximum_Length_Of_Process_Name_Value;

```

```

..// .....
..// MODULE NAME:
..//   process_managers_globals
..//
..// MODULE TYPE:
..//   package specification
..//
..// MODULE PURPOSE:
..//   an example of (and a template for) an instantiation of the generic
..//   package generic_process_managers_globals
..//
..// .....
..// MODULE DESCRIPTION:
..//   package instantiating the process managers globals information where:
..//   maximum_length_of_process_name_value is defined as 32 characters
..//
..// MODULE CONTENTS:
..//   process_managers_globals (package specification)
..//   device_name_type (type)
..//   how to handle message queue overflow (type)
..//   maximum_length_of_process_name (constant)
..//   process_name_type (type)
..//
..// REFERENCES:
..//
..// DESIGN DOCUMENTS:
..//   Kernel Facilities Definition
..//
..// USER'S MANUAL:
..//   Kernel User's Manual
..//
..// TESTING AND VALIDATION:
..//   none
..//
..// NOTES:
..//   none
..//
..// .....
..// MODIFICATION HISTORY:
..//   18aug88 bamberg created (complies clean)
..//   25aug89 bamberg final doc. check for 01sep89 freeze
..//
..// .....
..// VERSION: 3.0
..//
..// DISCLAIMER:
..//
..// The following copyright must be included in this software and
..// all software utilizing this software.
..//
..// DISTRIBUTION:
..//
..// Approved for public release; distribution is unlimited
..//
..// Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
..// PA. The Software Engineering Institute (SEI) is a federally

```

```

..// funded research and development center established and operated
..// by Carnegie Mellon University (CMU). Sponsored by the U.S.
..// Department of Defense under contract F19628-85-C-0003, the SEI
..// is supported by the services and defense agencies, with the U.S.
..// Air Force as the executive contracting agent.
..//
..// Permission to use, copy, modify, or distribute this software and
..// its documentation for any purpose and without fee is hereby
..// granted, provided that the above copyright notice appear in all
..// copies and that both that copyright notice and this permission
..// notice appear in supporting documentation. Further, the names
..// Software Engineering Institute or Carnegie Mellon University may
..// not be used in advertising or publicity pertaining to
..// distribution of the software without specific, written prior
..// permission. CMU makes no claims or representations about the
..// suitability of this software for any purpose. This software is
..// provided "as is" and no warranty, express or implied, is made
..// by the SEI or CMU, as to the accuracy and functioning of the
..// program and related program material, nor shall the fact of
..// distribution constitute any such warranty. No responsibility is
..// assumed by the SEI or CMU in connection herewith.
..// .....
with Generic Process Managers Globals;
pragma Elaborate (Generic_Process_Managers_Globals);
.. to get: see MODULE CONTENTS
..// .....
..// package short name: PMG
..// .....
package Process_Managers_Globals
is new Generic Process Managers Globals (
  Maximum_Length_Of_Process_Name_Value => 32);
pragma (page);

```



```

--/ .....
--/ MODULE NAME:
--/   generic_process_managers
--/
--/ MODULE TYPE:
--/   generic_package_specification
--/
--/ MODULE PURPOSE:
--/   support the declaration and creation of the logical processor
--/   configuration (i.e., the Process Table)
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   collection of subprograms to declare and create the logical processor
--/   configuration
--/
--/ MODULE CONTENTS:
--/   generic_process_managers (generic_package_specification)
--/   calling_unit_not_Main_Unit (exception)
--/   create_process (procedure)
--/   declare_process (overloaded [2] function)
--/   illegal_process_address_enabled (generic_formal_parameter)
--/   illegal_process_identifier_enabled (generic_formal_parameter)
--/   illegal_process_identifier_enabled (generic_formal_parameter)
--/   insufficient_space (exception)
--/   insufficient_space_enabled (generic_formal_parameter)
--/   maximum_message_queue_size_value (generic_formal_parameter)
--/   maximum_process_stack_value (generic_formal_parameter)
--/   no_kernel_process_on_non_kernel_device (exception)
--/   no_kernel_process_on_non_kernel_device_enabled
--/       (generic_formal_parameter)
--/   process_already_created (exception)
--/   process_already_created_enabled (generic_formal_parameter)
--/   process_already_exists (exception)
--/   process_already_exists_enabled (generic_formal_parameter)
--/   unknown_non_kernel_device (exception)
--/   unknown_non_kernel_device_enabled (generic_formal_parameter)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/   Kernel Facilities Definition
--/   --REQ: 5.1.6; 5.1.8; 6.1.9; 7.1.1; 7.1.2;
--/   --REQ: 7.1.8; 7.1.11; 7.1.13
--/   --PRIM: 16.1.1; 16.1.2
--/   --TIME: N/A
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Unit testing
--/
--/ NOTES:

```

```

--/ In declare_process:
--/   If checking for calling_unit is not Main_Unit is disabled, the
--/   application is asserting that it guarantees the calling unit is the
--/   Main_Unit (except for Child of DARK, when this restriction may be
--/   lifted); if the calling unit is NOT the Main_Unit and checking is
--/   disabled, currently, a null process ID is returned (this is
--/   required to maintain the integrity of the Kernel)
--/
--/ In create_process:
--/   If checking for calling_unit is not Main_Unit is disabled, the
--/   application is asserting that it guarantees the calling unit is the
--/   Main_Unit (except for Child of DARK, when this restriction may be
--/   lifted); if the calling unit is NOT the Main_Unit and checking is
--/   disabled, currently, this procedure simply returns (this is
--/   required to maintain the integrity of the Kernel)
--/
--/   If checking for illegal_process_address is disabled, the application is
--/   asserting that it guarantees the input process address is good;
--/   however, since the Kernel cannot sufficiently protect itself if this
--/   address is really a bad address, it must do the check in all cases;
--/   if the address is truly a bad address, the Kernel assigns the address
--/   of a null procedure to be process address; the process will then do
--/   nothing, but do so without affecting the integrity of the Kernel
--/
--/   If checking for illegal_process_identifier is disabled, the application
--/   is asserting that the input process_ID was, in fact, created via a
--/   prior call to declare_process; if that is not the case, the input
--/   value is some null value (null or null_process), and the call simply
--/   returns, or it is
--/   assumed that somewhere within the rest of create_process, something
--/   is bound to cause an exceptional exit (thus maintaining the integrity
--/   of the Kernel); this latter case is deemed to be "OK" because the
--/   only way this
--/   could happen is if the application used unchecked programming to
--/   convert a non-process_identifier value into a process_ID)
--/
--/ In declare_process:
--/   If checking for insufficient_space is disabled, the application is
--/   asserting that there is sufficient storage to allocate the data
--/   structures for the process table entry; if this is not the case and
--/   checking is disabled, a null value is returned
--/
--/ In create_process:
--/   If checking for insufficient_space is disabled, the application is
--/   asserting that there is sufficient space for:
--/   the process stack; if this is not the case and checking is
--/   disabled, the Kernel assigns the address of a null procedure to
--/   be the process address; the process will then do nothing, but
--/   do so without affecting the integrity of the Kernel
--/   the message queue; if this is not the case and checking is
--/   disabled, the Kernel assigns the address of a null procedure to
--/   be the process address; the process will then do nothing, but
--/   do so without affecting the integrity of the Kernel
--/   and that there is no overflow (cycling around) of process numbers; if
--/   overflow occurs and checking is disabled, this procedure simply
--/   returns (this is required to maintain the integrity of the Kernel)
--/
--/   If checking for no_kernel_process_on_non_kernel_device is disabled, the
--/   application is asserting that the process to be created is a Kernel
--/   process on a Kernel processor; if this is not the case and checking is
--/   disabled, this procedure simply returns (this is required to maintain

```

```

--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
pragma (page);

```

```

--/ the integrity of the kernel)
--/ If checking for process already created is disabled, the application is
--/ asserting that that no multiple calls upon create_process are made
--/ with a given process ID; if such calls are made, nothing is done
--/ (i.e., an immediate return); the original creation stands
--/ If checking for process already exists is disabled, the application is
--/ asserting that it guarantees unique process logical names; otherwise,
--/ if there is a duplicate name, as it causes the kernel no internal
--/ problems, no checking is done; however the application may see some
--/ unexpected outputs (via tool interface, name_of)
--/ If checking for unknown non_kernel_device is disabled, the application
--/ is asserting that it guarantees that the declaration of the non-kernel
--/ device name matches with a processor logical name in the NCT; if it
--/ does not, the kernel flags this as being an undefined remote process
--/ (i.e., the kernel cannot determine the process's index) during
--/ initialization, complete processing, thus maintaining
--/ kernel integrity
--/
--/ .....

```

MODIFICATION HISTORY:

```

--/ 18aug88 bamberg created (compiles clean)
--/ 27aug89 bamberg final doc. check for 01sep89 freeze
--/
--/ .....

```

```

--/ VERSION: 3.0
--/
--/ .....

```

DISCLAIMER:

```

--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ .....

```

DISTRIBUTION:

```

--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ .....

```

```

--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the

```

```

with Kernel Exceptions;
.. to get calling unit not Main Unit
.. illegal process address
.. illegal process identifier
.. insufficient space
.. no Kernel process on non_Kernel_device
.. process already created
.. process already exists
.. unknown non_Kernel_device

with Hardware Interface; use Hardware_Interface;
.. to get: hw address
.. hw long positive
.. hw long natural

with Process Managers Globals;
.. to get: device_name type
.. how to handle message queue overflow
.. process_name_type

with Process_Table;
.. to get: process identifier

with Schedule_Types;
.. to get: default_preemption
.. default priority
.. preemption
.. priority

pragma (page);

generic
.. ERROR checking conditional compilation flags
.. REQ: 5.1.8

Calling Unit Not Main Unit Enabled: in Boolean := True;
Illegal_Process_Address_Enabled: in Boolean := True;
Illegal_Process_Identifier_Enabled: in Boolean := True;
Insufficient_Space_Enabled: in Boolean := True;
No_Kernel_Process_On_Non_Kernel_Device_Enabled: in Boolean := True;
Process_Already_Created_Enabled: in Boolean := True;
Process_Already_Exists_Enabled: in Boolean := True;
Unknown_Non_Kernel_Device_Enabled: in Boolean := True;

.. value indicating the maximum size of the incoming message queue in number
.. of bytes (this value is never used within the Kernel except for the
.. initialization of the size of the message queue, nor should it be used
.. without the Kernel)
.. REQ: 5.1.6; (7.1.11)

Maximum_Message_Queue_Size_Value: in Hw_Long_Natural;

.. value indicating the maximum size of the process stack in number of
.. bytes (this value is never used within the Kernel except for the
.. initialization of the size of the process stack, nor should it be used
.. without the Kernel)
.. REQ: 5.1.6; 7.1.8

Maximum_Process_Stack_Size_Value: in Hw_Long_Positive;
Pragma(Page);
.. package short name: GPM
.. .....

package Generic_Process_Managers
is
.. REQ: 6.1.9; 7.1.1; 7.1.2; 7.1.13
.. calling unit not Main Unit raised by:
.. declare process
.. create process
.. PRIM: 16.1.1; 16.1.2

Calling Unit Not Main Unit: exception
renames Kernel_Exceptions.Calling_Unit_Not_Main_Unit;

```

```

.. Illegal process address raised by:
..   create_process
..
.. PRIM: 16.1.2
Illegal Process Address : exception
renamese Kernel_Exceptions.Illegal_Process_Address;

.. Illegal process identifier raised by:
..   create_process
..
.. PRIM: 16.1.2
Illegal Process Identifier : exception
renamese Kernel_Exceptions.Illegal_Process_Identifier;

.. Insufficient space raised by:
..   declare_process
..   create_process
..
.. PRIM: 16.1.1; 16.1.2
Insufficient Space : exception
renamese Kernel_Exceptions.Insufficient_Space;

.. no Kernel process on non_Kernel_device raised by:
..   create_process
..
.. PRIM: 16.1.2
No Kernel Process On Non Kernel Device : exception
renamese Kernel_Exceptions.No_Kernel_Process_On_Non_Kernel_Device;

.. process already created raised by:
..   create_process
..
.. PRIM: 16.1.2
Process Already Created : exception
renamese Kernel_Exceptions.Process_Already_Created;

.. process already exists raised by:
..   declare_process
..
.. PRIM: 16.1.1
Process Already Exists : exception
renamese Kernel_Exceptions.Process_Already_Exists;

```

```

.. unknown non_Kernel_device raised by:
..   declare_process
..
.. PRIM: 16.1.1
Unknown Non Kernel Device : exception
renamese Kernel_Exceptions.Unknown_Non_Kernel_Device;

pragma (page);

```

```

function Declare_Process (
  Name : Process_Managers_Globals.Process_Name_Type;
  return Process_Table.Process_Identifier;
--/
--/ SUBPROGRAM NAME:
--/ declare_process
--/
--/ DESCRIPTION:
--/ declares all remote Kernel processes with which communication may
--/ occur, and any local processes to be created on this processor;
--/ associates a logical user-defined "name" (string valued) with a
--/ process ID (internally generated handle) and initialized Kernel data
--/ structures
--/
--/ -REQ: 7.1.3, 10.1.1; 10.1.6, 10.1.15
--/
--/ -PRIM: 16.1.1
--/
--/ --TIME: 6.2.1
--/
--/ PARAMETERS:
--/ name
--/ user-defined string value by which the process is referred in the
--/ user's problem domain space
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the process ID; the ID (handle) by which this process is
--/ referenced in all other Kernel calls
--/
--/ default value:
--/ none
--/
--/ ASSUMPTIONS:
--/ prior network initialization sequence completed successfully (i.e.,
--/ calls to initialize_processor and start_subordinates as
--/ appropriate)
--/
--/ PRECONDITIONS:
--/ the calling unit is the Ada Main Unit
--/ the user-specified name has not been declared by a prior
--/ invocation of declare_process
--/
--/ ACTIONS:
--/ the Kernel initializes the process information for the process
--/ the Kernel generates a processor-unique process ID - the handle by
--/ which this process is referenced for all ensuing activities - and
--/ returns that handle to the user
--/
--/ POSTCONDITIONS:
--/ updated Kernel data structures
--/ processor-unique process ID returned to the user
--/

```

```

--/ ERROR PROCESSING:
--/ when the call is detected from a unit that is not the Main Unit:
--/ the Kernel raises calling_unit_not_Main_Unit_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/ when there is insufficient memory in which to construct internal
--/ Kernel data structures:
--/ the Kernel raises insufficient_space_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/ when the value for the name parameter has appeared in a prior
--/ declare_process call:
--/ the Kernel raises process_already_exists_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/
--/
--/ =====
pragma (page);

```

```

function Declare_Process (
  Name : Process_Managers_Globals.Device_Name_Type)
  return Process_Table.Process_Identifier;
--/
--/
--/ SUBPROGRAM NAME:
--/ declare process
--/
--/ DESCRIPTION:
--/ declares any remote, non-Kernel devices with which communication may
--/ occur; associates a logical, user-defined "name" (string-valued)
--/ with a process ID (internally generated handle) and initialized
--/ Kernel data structures
--/
--/ --REQ: 7.1.3; 10.1.1; 10.1.6; 10.1.15; 10.1.26
--/
--/ --PRIM: 16.1.1
--/
--/ --TIME: 6.2.1
--/
--/ PARAMETERS:
--/ name
--/ user-defined string value by which the non-Kernel device is
--/ referred in the user's problem domain space
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the process ID; the ID (handle) by which this process is
--/ referenced in all other Kernel calls
--/
--/ default value:
--/ none
--/
--/ ASSUMPTIONS:
--/ prior network initialization sequence completed successfully (i.e.,
--/ calls to Initialize "_processor" and start_subordinates as
--/ appropriate)
--/
--/ PRECONDITIONS:
--/ the calling unit is the Ada Main Unit
--/ the indicated non-Kernel device is a valid non-Kernel device (i.e.,
--/ has been identified in the NCT)
--/ the user-specified name has not been declared by a prior
--/ invocation of declare_process
--/
--/ ACTIONS:
--/ the Kernel initializes the process information for the process
--/ the Kernel generates a processor unique process ID - the handle by
--/ which this process is referenced for all ensuing activities - and
--/ returns that handle to the user
--/
--/ POSTCONDITIONS:
--/ updated Kernel data structures
--/ processor-unique process ID returned to the user
--/
--/
--/ ERROR PROCESSING:
--/ when the call is detected from a unit that is not the Main Unit:
--/ the Kernel raises calling_unit_not_Main_Unit_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/ when there is insufficient memory in which to construct internal
--/ Kernel data structures:
--/ the Kernel raises insufficient_space_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/ when the value for the name parameter has appeared in a prior
--/ declare_process call:
--/ the Kernel raises process_already_exists_exception
--/
--/ POSTCONDITIONS:
--/ the process is not declared
--/ when the communication partner being declared is a non-Kernel device
--/ and that non-Kernel device is not known as such (i.e., there is no
--/ entry for it in the NCT):
--/ the Kernel raises unknown_non_Kernel_device
--/
--/ POSTCONDITIONS:
--/ the communication partner is not declared
--/
--/
--/ =====
pragma (page);

```

```

procedure Create_Process (
  Process_ID : in Process_Table.Process_Identifier;
  Address : in Hw_Address;
  Stack_Size : in Hw_Long_Positive := Maximum_Process_Stack_Size_Value;
  Message_Queue_Size : in Hw_Long_Natural :=
    Maximum_Message_Queue_Size_Value;
  Message_Queue_Overflow_Handling : in
    Process_Managers_Globals.Drop_Newest_Message;
  Initial_Priority : in Schedule_Type.Priority :=
    Schedule_Type.Default_Priority;
  Preemptible : in Schedule_Type.Preemption :=
    Schedule_Type.Default_Preemption;
  SUBPROGRAM NAME:
  create_process
  --
  -- DESCRIPTION:
  -- creates an independent thread of control; creates the execution
  -- environment in which the thread is to execute; this thread is called
  -- a process
  --
  -- REQ: 7.1.4; 7.1.5; 7.1.6; 7.1.7; 7.1.8; 7.1.9; 7.1.10
  -- REQ: 7.1.11; 7.1.12; 7.1.25; 7.1.26; 7.1.27
  -- REQ: 9.1.1; 9.1.2; 9.1.7; 9.1.11; 10.1.22; 10.1.23
  --
  -- PRIM: 16.1.2
  --
  -- TIME: 6.2.1; 7.2.1; 7.2.5; 7.2.6
  --
  PARAMETERS:
  process_ID
  the handle by which this process is referenced; generated by a
  previous call to declare_process
  default value:
  none
  address
  a value of hw_address that corresponds to the 'address attribute
  of the procedure being identified as a Kernel process
  default value:
  none
  stack_size
  the maximum number of bytes reserved for process-local storage
  (i.e., storage for the non-Kernel portion of the runtime stack
  for the process - as generated by the Ada compiler)
  default value:
  maximum_process_stack_size_value, a user-specified maximum
  message_queue_size
  the maximum number of bytes reserved for incoming messages to this

```

```

-- process from all other processes
--
-- default value:
-- maximum_message_queue_size_value, a user-specified maximum
--
-- message_queue_overflow_handling
-- an indication of how overflow of the incoming message queue is to
-- be handled for this process
--
-- values include:
-- drop_newest_message (no change to the contents of the message
-- queue; messages existing in message queue are preserved;
-- the newest message is simply dropped)
-- others TBD (thus, the only option right now is to drop the
-- newest message)
--
-- default value:
-- drop_newest_message (messages existing in message queue are
-- preserved)
--
-- Initial priority
-- the priority at which this process is initially registered with
-- the Scheduler
--
-- default value:
-- the Kernel-defined default_priority (the lowest priority
-- available)
--
-- preemptible
-- indication of whether or not a process may be preempted by
-- another process of the same priority
--
-- values include:
-- enabled (the process is preemptable)
-- disabled (the process is not preemptable)
--
-- default value:
-- the Kernel-defined default_preemption (enabled; preemptable)
--
-- ASSUMPTIONS:
-- all parameters have legal values
--
-- PRECONDITIONS:
-- declare_process was previously called to create a legal process_ID
-- calling unit is the Ada Main Unit
-- this Kernel process is being created on a Kernel processor
-- this process is not already created
-- the address provided is a legal address
-- there is sufficient space to create this process and its execution
-- environment
--
-- ACTIONS:
-- updates Kernel data structures, indicating the creation of this
-- process
-- constructs the execution environment for this process, including:

```

```

--/ the local process stack (stack plug, dummy call frame, space or
--/ local process variables); and the process control block (message
--/ queue, schedule attributes, process code, and context save area
--/ pointers)
--/ the local Kernel informs all other Kernels about the creation of this
--/ process
--/
--/ POSTCONDITIONS:
--/ Kernel data structures updated to reflect the creation of this
--/ process
--/ Initial process characteristics recorded in appropriate internal data
--/ structures
--/ execution environment created for this process
--/ this process is registered as "suspended" with the Kernel Scheduler
--/ and is queued with all other processes of the same priority level
--/ to begin execution at the discretion of the Scheduler
--/ all other Kernels informed about the existence of this process
--/
--/ ERROR PROCESSING:
--/ when the call is detected from a unit that is not the Main Unit:
--/ the Kernel raises calling_unit_not_Main_Unit_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/ when the address can be determined to be invalid:
--/ the Kernel raises illegal_process_address_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/ when the process ID cannot be found within Kernel data structures
--/ or is a null value:
--/ the Kernel raises illegal_process_identifier_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/ when there is insufficient memory in which to construct the execution
--/ environment for the process being created (process number wrap-around,
--/ not enough room for the process stack or the incoming message queue):
--/ the Kernel raises insufficient_space_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/ when the creation of a Kernel process on a non-Kernel device is
--/ detected:
--/ the Kernel raises no_Kernel_process_on_non_Kernel_device_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/ when a process has previously been created for this process_ID value:
--/ the Kernel raises process_already_created_exception
--/ POSTCONDITIONS:
--/ the process is not created
--/
--/ end Generic Process Managers;
--/ pragma (page);

```

```

--/ .....
--/ MODULE NAME:
--/ process_managers
--/
--/ MODULE TYPE:
--/ package specification
--/
--/ MODULE PURPOSE:
--/ an example of (and a template for) an instantiation of the generic
--/ package generic_process_managers
--/
--/ MODULE DESCRIPTION:
--/ package instantiating the declaration and creation of the logical
--/ processor configuration capabilities where:
--/ error checking for calling_unit_not_Main_Unit is enabled
--/ error checking for illegal_process_address is enabled
--/ error checking for illegal_process_identifier is enabled
--/ error checking for insufficient_space is enabled
--/ error checking for no_Kernel_process_on_non_Kernel_device is enabled
--/ error checking for process_already_exists is enabled
--/ error checking for unknown_non_Kernel_device is enabled
--/ maximum_message_queue_size_value is defined as 1024
--/ maximum_process_stack_size_value is defined as 4096
--/
--/ MODULE CONTENTS:
--/ process_managers (package specification)
--/ calling_unit_not_Main_Unit_exception
--/ create_process (procedure)
--/ declare_process (overloaded (2) function)
--/ illegal_process_address_exception
--/ illegal_process_identifier_exception
--/ insufficient_space_exception
--/ no_Kernel_process_on_non_Kernel_device_exception
--/ process_already_created_exception
--/ process_already_exists_exception
--/ unknown_non_Kernel_device_exception
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ none
--/
--/ USER'S MANUAL:
--/ Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/ none
--/
--/ NOTES:
--/ none
--/
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (complies clean)

```



```

../ 27aug89 bernberg    final doc. check for 01sep89 freeze
../
../-----
../ VERSION: 3.0
../
../ DISCLAIMER:
../
../ The following copyright must be included in this software and
../ all software utilizing this software.
../
../ DISTRIBUTION:
../
../ Approved for public release, distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ of Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../ .....
with Generic Process Managers;
pragma Elaborate (Generic Process Managers);
.. to get: see MODULE CONTENTS
../ .....
../ package short name: PM
../ .....
package Process Managers

```

```

is new Generic Process Managers (
Calling Unit Not Main Unit Enabled => True,
Illegal Process Address Enabled => True,
Illegal Process Identifier Enabled => True,
Insufficient Space Enabled => True,
No Kernel Process On Non_Kernel_Device Enabled => True,
Process Already Created Enabled => True,
Process Already Exists Enabled => True,
Unknown Non_Kernel_Device Enabled => True,
Maximum Message Queue Size Value => 1024,
Maximum Process Stack Size Value => 4096);
pragma (page);

```

```

./ .....
./ MODULE NAME:
./   generic_communication_globals
./
./ MODULE TYPE:
./   generic_package_specification
./
./ MODULE PURPOSE:
./   exports communication global information
./
./ .....
./ MODULE DESCRIPTION:
./   collection of globals related to communication management but which
./   can be used generally by anyone
./
./ MODULE CONTENTS:
./   generic_communication_globals (generic package specification)
./   maximum_message_length (constant)
./   maximum_message_length_value (generic formal parameter)
./   message_length_type (type)
./   message_tag_type (type)
./
./ REFERENCES:
./   DESIGN DOCUMENTS:
./     Kernel Facilities Definition
./     --REQ: 5.1.4, 5.1.7
./     --PRIM: N/A
./     --TIME: N/A
./
./   USER'S MANUAL:
./     Kernel User's Manual
./
./ TESTING AND VALIDATION:
./   Integration testing with generic_communication_management
./
./ NOTES:
./   none
./
./ .....
./ MODIFICATION HISTORY:
./   18aug88 barnberg created (complies clean)
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./
./ The following copyright must be included in this software and
./ all software utilizing this software.
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,

```

```

./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....

```

pragma (page);

```

with Hardware Interface; use Hardware_Interface;
-- to get: hw_Integer
-- hw_natural

generic
-- the maximum length of a message that can be sent across the network
Maximum_Message_Length_Value : in Hw_Natural;

-- *****
-- package short name: GCG
-- *****

package Generic_Communication_Globals

is
-- the maximum length of a message that can be sent across the network
Maximum_Message_Length : constant Hw_Natural :=
Maximum_Message_Length_Value;

-- this type defines the limits of message length
type Message_Length_Type is new Hw_Natural;

-- this type defines the limits of message tags
type Message_Tag_Type is new Hw_Integer;

and Generic_Communication_Globals;
pragma page;

-- *****
-- MODULE NAME:
-- communication_globals
--
-- MODULE TYPE:
-- package specification
--
-- MODULE PURPOSE:
-- an example of (and a template for) an instantiation of the generic
-- package generic_communication_globals
--
-- *****
-- MODULE DESCRIPTION:
-- package instantiating the global communication information where:
-- maximum_message_length value is defined as 1024
--
-- MODULE CONTENTS:
-- communication_globals (package specification)
-- maximum_message_length (constant)
-- message_length_type (type)
-- message_tag_type (type)
--
-- REFERENCES:
-- DESIGN DOCUMENTS:
-- none
--
-- USER'S MANUAL:
-- Kernel User's Manual
--
-- TESTING AND VALIDATION:
-- none
--
-- NOTES:
-- none
--
-- *****
-- MODIFICATION HISTORY:
-- 18aug88 bamborg created (complies clean)
-- 25aug88 bamborg final doc. check for 01sep89 freeze
--
-- *****
-- VERSION: 3.0
--
-- DISCLAIMER:
--
-- The following copyright must be included in this software and
-- all software utilizing this software
--
-- DISTRIBUTION:
--
-- Approved for public release; distribution is unlimited.
--
-- Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
-- PA. The Software Engineering Institute (SEI) is a federally
-- funded research and development center established and operated

```

.. by Carnegie Mellon University (CMU). Sponsored by the U.S.
 .. Department of Defense under contract F19628-85-C-0003, the SEI
 .. is supported by the services and defense agencies, with the U.S.
 .. Air Force as the executive contracting agent.

.. Permission to use, copy, modify, or distribute this software and
 .. its documentation for any purpose and without fee is hereby
 .. granted, provided that the above copyright notice appear in all
 .. copies and that both that copyright notice and this permission
 .. notice appear in supporting documentation. Further, the names
 .. Software Engineering Institute or Carnegie Mellon University may
 .. not be used in advertising or publicity pertaining to
 .. distribution of the software without specific, written prior
 .. permission. CMU makes no claims or representations about the
 .. suitability of this software for any purpose. This software is
 .. provided "as is" and no warranty, express or implied, is made
 .. by the SEI or CMU, as to the accuracy and functioning of the
 .. program and related program material, nor shall the fact of
 .. distribution constitute any such warranty. No responsibility is
 .. assumed by the SEI or CMU in connection herewith.

with Generic Communication Globals;
 pragma Elaborate (Generic Communication Globals);
 .. to get: see MODULE CONTENTS

.. ..
 .. package short name CG

package Communication Globals

is new Generic Communication Globals {
 Maximum Message Length_Value => 1024};
 pragma (page);

.. ..
 .. MODULE NAME:
 .. generic_communication_management
 ..
 .. MODULE TYPE:
 .. generic package specification
 ..
 .. MODULE PURPOSE:
 .. support communication among Kernel processes
 ..
 ..
 .. MODULE DESCRIPTION:
 .. collection of subprograms to send and receive messages among processes
 ..
 .. MODULE CONTENTS:
 .. generic_communication_management (generic package specification)
 .. allocate_device_receiver (procedure)
 .. buffer too small for message enabled (generic formal parameter)
 .. illegal context for call (exception)
 .. illegal context for call enabled (generic formal parameter)
 .. illegal process identifier (exception)
 .. illegal process identifier enabled (generic formal parameter)
 .. message not received (exception)
 .. message not received enabled (generic formal parameter)
 .. message timed out (exception)
 .. message timed out enabled (generic formal parameter)
 .. network failure (exception)
 .. network failure enabled (generic formal parameter)
 .. no message available (exception)
 .. no message available enabled (generic formal parameter)
 .. no such device exists (exception)
 .. no such device exists enabled (generic formal parameter)
 .. receive message (overloaded (3) procedure)
 .. receiver dead (exception)
 .. receiver dead enabled (generic formal parameter)
 .. receiver is sender (exception)
 .. receiver is sender enabled (generic formal parameter)
 .. receiver never existed (exception)
 .. receiver never existed enabled (generic formal parameter)
 .. replacing previous allocation (exception)
 .. replacing previous allocation enabled (generic formal parameter)
 .. send message (procedure)
 .. send message and wait (overloaded (3) procedure)
 ..
 .. REFERENCES:
 .. DESIGN DOCUMENTS:
 .. Kernel Facilities Definition
 .. --REQ: 6.1.9, 6.1.11; 10.1.19; 10.1.22; 10.1.23;
 .. --REQ: 10.1.28; 10.1.29; 10.1.30; 10.1.31; 10.1.33, 10.1.34
 .. --PRIM: N/A
 .. --TIME: N/A
 ..
 .. USER'S MANUAL:
 .. Kernel User's Manual

.....
pragma (page);

.. / TESTING AND VALIDATION:
.. / Unit testing
.. /
.. / NOTES:
.. / In send message send message and wait and allocate device receiver:
.. / if illegal process identifier checking is disabled, if the process
.. / identifier is illegal, the application will not be notified of this
.. / fact, and no action (sending or allocation) will occur.
.. /
.. / MODIFICATION HISTORY:
.. / 18aug88 bamborg created (compiles clean)
.. / 16sep88 dvk changed to use package short names
.. / 31mar89 tac Bug 0021 SEI: change send * procedure to have
.. / default parameters for length and address.
.. / 09may89 tac Bug 0021 SEI: update comments
.. / 29jun89 dvk Bug 0040 SEI: add checks for illegal PIDs
.. / 26aug89 tac minor comment changes to update req comments
.. /
.. / VERSION: 3.0
.. /
.. / DISCLAIMER:
.. / The following copyright must be included in this software and
.. / all software utilizing this software.
.. / DISTRIBUTION:
.. / Approved for public release; distribution is unlimited.
.. /
.. / Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
.. / PA. The Software Engineering Institute (SEI) is a federally
.. / funded research and development center established and operated
.. / by Carnegie Mellon University (CMU). Sponsored by the U.S.
.. / Department of Defense under contract F19628-85-C-0003, the SEI
.. / is supported by the services and defense agencies, with the U.S.
.. / Air Force as the executive contracting agent.
.. /
.. / Permission to use, copy, modify, or distribute this software and
.. / its documentation for any purpose and without fee is hereby
.. / granted, provided that the above copyright notice appear in all
.. / copies and that both that copyright notice and this permission
.. / notice appear in supporting documentation. Further, the names
.. / of Software Engineering Institute or Carnegie Mellon University may
.. / not be used in advertising or publicity pertaining to
.. / distribution of the software without specific, written prior
.. / permission. CMU makes no claims or representations about the
.. / suitability of this software for any purpose. This software is
.. / provided "as is" and no warranty, express or implied, is made
.. / by the SEI or CMU, as to the accuracy and functioning of the
.. / program and related program material, nor shall the fact of
.. / distribution constitute any such warranty. No responsibility is
.. / assumed by the SEI or CMU in connection herewith.

```

with Kernel_Exceptions;
.. to get: buffer too small for message
.. illegal context for call
.. illegal process identifier
.. message not received
.. message timed out
.. network failure
.. no message available
.. no such device exists
.. receiver dead
.. receiver is sender
.. receiver never existed
.. replacing_previous_allocation

with Hardware_Interface; use Hardware_Interface;
.. to get: hw address
.. null_hw_address

with Communication_Globals;
.. to get: message_length_type
.. message_lag_type

with Network_Globals;
.. to get: processor_identifier

with Process_Table;
.. to get: process_identifier

with Schedule_Types;
.. to get: current_process_priority
.. priority

with Time_Globals;
.. to get: elapsed_time
.. epoch_time
.. to kernel_time

generic

.. ERROR checking conditional compilation flags
..
..REQ: 5.1.8

Buffer Too Small For Message Enabled: In Boolean := True;
Illegal Context For Call Enabled: In Boolean := True;
Illegal Process Identifier Enabled: In Boolean := True;
Message Not Received Enabled: In Boolean := True;
Message Timed Out Enabled: In Boolean := True;
Network Failure Enabled: In Boolean := True;
No Message Available Enabled: In Boolean := True;
No Such Device Exists Enabled: In Boolean := True;
Receiver Dead Enabled: In Boolean := True;
Receiver Is Sender Enabled: In Boolean := True;

Receiver Never Existed Enabled: In Boolean := True;
Replacing_Previous_Allocation_Enabled: In Boolean := True;
../
../package short name: GCM
../
package Generic_Communication_Management
is
package Ke
renames Kernel_Exceptions;
package Cg
renames Communication_Globals;
package Ng
renames Network_Globals;
package Pt
renames Process_Table;
package St
renames Schedule_Types;
package Tg
renames Time_Globals;

REQ: 6.1.9; 6.1.11;
REQ: 10.1.19; 10.1.22; 10.1.23; 10.1.28; 10.1.29; 10.1.30; 10.1.31

.. buffer too small for message raised by:
.. receive message
.. receive message (waiting for)
.. receive message (waiting until)
..PRIM: 19.1.3

Buffer Too Small For Message: exception
renames Ke.Buffer_Too_Small_For_Message;

.. illegal context for call raised by:
.. send message and wait
.. send message and wait (for)
.. send message and wait (until)
.. receive message
.. receive message (waiting for)
.. receive message (waiting until)
..PRIM: 19.1.2; 19.1.3

```

Illegal Context For Call : exception
renames Ke.Illegal_Context_For_Call;

-- illegal_process_identifier_raised by:
-- send_message
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)
-- allocate_device_receiver

--PRIM: 10.1.31

Illegal Process Identifier : exception
renames Ke.Illegal_Process_Identifier;

-- message_not_received_raised by:
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)

--PRIM: 10.1.2

Message Not Received : exception
renames Ke.Message_Not_Received;

-- message_timed_out_raised by:
-- send_message_and_wait (for)
-- send_message_and_wait (until)
-- receive_message (waiting for)
-- receive_message (waiting until)

--PRIM: 10.1.2; 10.1.3

Message Timed Out : exception
renames Ke.Message_Timed_Out;

-- network_failure_raised by:
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)

--PRIM: 10.1.2

Network Failure : exception
renames Ke.Network_Failure;

-- no_message_available_raised by:
-- receive_message (waiting for)
-- receive_message (waiting until)

--PRIM: 10.1.3

No Message Available : exception
renames Ke.No_Message_Available;

-- no_such_device_exists_raised by:
-- allocate_device_receiver

--PRIM: 10.1.4

No Such Device Exists : exception
renames Ke.No_Such_Device_Exists;

-- receiver_dead_raised by:
-- send_message
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)

--PRIM: 10.1.1; 10.1.2

Receiver Dead : exception
renames Ke.Receiver_Dead;

-- receiver_is_sender_raised by:
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)

--PRIM: 10.1.2

Receiver Is Sender : exception
renames Ke.Receiver_Is_Sender;

-- receiver_never_existed_raised by:
-- send_message
-- send_message_and_wait
-- send_message_and_wait (for)
-- send_message_and_wait (until)

--PRIM: 10.1.1; 10.1.2

```
Receiver Never Existed : exception
renames Ka Receiver_Never_Exist;

replacing_previous_allocation_raised_by:
allocate_device_receiver

PRIM: 16.1.4

Replacing_Previous_Allocation : exception
renames Ke Replacing_Previous_Allocation;
pragma (page);
```

```
.....
procedure Send_Message ;
Receiver : In Pib.Process_Identifier;
Message_Tag : In Cg.Message_Tag_Type;
Message_Length : In Cg.Message_Length_Type := 0 ;
Message_Text : In Hw.Address := Null_Hw_Address ;
--/ =====
--/ SUBPROGRAM NAME:
--/ send_message
--/
--/ DESCRIPTION:
--/ does an asynchronous send of a message to a specified receiver;
--/ there is no waiting for any kind of acknowledgement by the receiver
--/
--/ REQ: 7.1.21; 7.1.22; 10.1.2; 10.1.5; 10.1.6; 10.1.25; 10.1.26
--/ REQ: 14.1.5
--/
--/ PRIM: 10.1.31; 19.1.1
--/
--/ TIME: 10.2.1; 10.2.2; 10.2.5; 10.2.6; 10.2.11
--/ TIME: 14.2.3, 14.2.5, 14.2.6
--/
--/ PARAMETERS:
--/ receiver
--/ the local process ID of the intended receiver of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing message-specific data
--/
--/ default value:
--/ 0
--/
--/ message_text
--/ the address at which the message-specific data begin; the Kernel
--/ collects message_length bytes of data beginning at the address
--/ specified by message_text
--/
--/ default value:
--/ null_hw_address
--/
--/ PRECONDITIONS:
--/ NONE
--/
```



```

--/ ACTIONS
--/ the Kernel validates the receiver field
--/ the Kernel sends the message from the caller to the specified
--/ receiver
--/
--/ POSTCONDITIONS:
--/ the message is sent to the intended receiver, with no guarantee of
--/ message receipt
--/
--/ ERROR PROCESSING:
--/ when the local Kernel detects that the specified receiver is dead:
--/ the Kernel raises receiver_dead exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ when the local Kernel has no knowledge of the specified receiver:
--/ the Kernel raises receiver_never_existed exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ when this primitive is called with an undeclared process ID
--/ the Kernel raises illegal_process_identifier exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/
--/
--/ pragma (page);

```

```

procedure Send_Message_And_Wait (
  Receiver : In Pib.Process_Identifier;
  Message_Tag : In Cg.Message_Tag_Type;
  Message_Length : In Cg.Message_Length_Type := 0;
  Message_Text : In Hw.Address := Null_Hw_Address;
  Resumption_Priority : In St.Priority := St.Current_Process_Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ send_message_and_wait
--/
--/ DESCRIPTION:
--/ does a synchronous send of a message to a specified receiver; the
--/ calling process blocks until a Kernel-to-Kernel acknowledgement of
--/ message receipt (or lack thereof) by the receiving process is
--/ returned
--/
--/ REQ: 7.1.21; 7.1.22; 7.1.25; 7.1.26
--/ REQ: 9.1.6; 9.1.8; 9.1.22; 10.1.2; 10.1.3
--/ REQ: 10.1.4; 10.1.5; 10.1.6
--/ REQ: 10.1.10; 10.1.11; 10.1.25; 10.1.27; 10.1.32
--/ REQ: 11.1.12; 14.1.5
--/
--/ PRIM: 10.1.31; 19.1.2
--/
--/ TIME: 10.2.3; 10.2.4; 10.2.7; 10.2.8; 10.2.11
--/ TIME: 14.2.3; 14.2.5; 14.2.6
--/
--/ PARAMETERS:
--/ receiver
--/ the process ID of the intended receiver of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing message-specific data
--/
--/ default value:
--/ 0
--/
--/ message_text
--/ the address at which the message-specific data begins; the Kernel
--/ collects message_length bytes of data beginning at the address
--/ specified by message_text
--/
--/ default value:
--/ null_hw_address

```

```

--/ resumption_priority
--/ the priority to be assigned to this process when the
--/ acknowledgement is received (i.e., the process becomes unblocked)
--/
--/ default value:
--/ the calling process's current priority
--/
--/ PRECONDITIONS:
--/ invoked from a non-blocking context
--/ the receiver is not the sender
--/
--/ ACTIONS:
--/ the Kernel validates the receiver field
--/ the Kernel indicates that a Kernel-to-Kernel acknowledgement is
--/ required when the receiver receives the message
--/ the Kernel sends the message from the caller to the specified
--/ receiver using local optimization if possible
--/ the calling process blocks until a Kernel-to-Kernel positive
--/ acknowledgement is returned, indicating that the message was
--/ received by the receiver
--/ the Kernel possibly changes the process priority
--/
--/ POSTCONDITION: 1.
--/ the message is sent to the intended receiver
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running depending
--/ on the discretion of the Kernel Scheduler
--/
--/ ERROR PROCESSING:
--/ when send_message_and_wait is called within an interrupt handler:
--/ the Kernel raises illegal_context_for_call
--/ POSTCONDITIONS:
--/ the call is rejected; no message is sent; no change in priority
--/ is registered
--/ when this primitive is called with an undeclared process ID
--/ the Kernel raises illegal_process_identifier_exception
--/ POSTCONDITIONS:
--/ the call is rejected; no message is sent; no change in priority
--/ is registered
--/ when the specified receiver of the message does not actually receive
--/ the message (e.g., the message is dropped or the process dies):
--/ the Kernel raises message_not_received_exception
--/ POSTCONDITIONS:
--/ the sent message is never received by the receiver
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when a network problem is detected:
--/ the Kernel raises network_failure_exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler

```

```

--/ when the local Kernel detects that the specified receiver is dead:
--/ the Kernel raises receiver_dead_exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when a process specifies itself as the intended receiver:
--/ the Kernel raises receiver_is_sender_exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when the local Kernel has no knowledge of the specified receiver:
--/ the Kernel raises receiver_never_existed_exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/
--/ pragma (page);

```

```

procedure Send Message And Wait (
  Receiver : In PibProcess Identifier;
  Message Tag : In CqMessage Tag Type;
  Message Length : In CqMessage Length Type := 0;
  Message Text : In Hw_Address := Null_Hw_Address;
  Timeout After : In Tg_Elapsed Time;
  Resumption Priority : In St.Priority := St.Current_Process_Priority);
--
SUBPROGRAM NAME:
-- send_message_and_wait
--
DESCRIPTION:
-- does a synchronous send of a message to a specified receiver; the
-- calling process blocks until a Kernel-to-Kernel acknowledgement of
-- message receipt (or lack thereof) by the receiving process is
-- returned or until the specified elapsed time has elapsed
--
--REQ: 7.1.21, 7.1.22, 7.1.25, 7.1.28
--REQ: 9.1.6, 9.1.8, 9.1.22, 10.1.2
--REQ: 10.1.3, 10.1.4, 10.1.5, 10.1.6, 10.1.7, 10.1.9,
--REQ: 10.1.10, 10.1.11, 10.1.25, 10.1.27, 10.1.32
--REQ: 11.1.12, 14.1.5
--
--PRIM: 10.1.31, 10.1.2
--
--TIME 10.2.3, 10.2.4, 10.2.7, 10.2.8, 10.2.11
--TIME: 14.2.3, 14.2.5, 14.2.8
--
PARAMETERS:
-- receiver
-- the process ID of the intended receiver of this message
--
-- default value:
-- none
--
-- message tag
-- an application-defined tag that allows the receiver to interpret
-- the message in the format the sender intended
--
-- default value:
-- none
--
-- message length
-- the number of bytes of text that should be interpreted as
-- containing message-specific data
--
-- default value:
-- 0
--
-- message text
-- the address at which the message-specific data begin; the Kernel
-- collects message length bytes of data beginning at the address
-- specified by message text
--
-- default value:

```

```

-- null_hw_address
--
-- timeout after
-- duration after which the attempt to send is abandoned and the
-- process becomes unblocked
--
-- default value:
-- none
--
-- resumption priority
-- the priority to be assigned to this process when the
-- acknowledgement is received (i.e., the process becomes unblocked)
--
-- default value:
-- the calling process's current priority
--
PRECONDITIONS:
-- invoked from a non-blocking context
-- the receiver is not the sender
--
ACTIONS:
-- the Kernel validates the receiver field
-- the Kernel indicates that a Kernel-to-Kernel acknowledgement is
-- required when the receiver receives the message
-- the Kernel bundles the elapsed time value with the message (i.e., the
-- timeout is a timeout at the receiver's site and measured with
-- respect to the remote processor's clock)
-- the Kernel sends the message from the caller to the specified
-- receiver using local optimization if possible
-- the calling process blocks until a Kernel-to-Kernel positive
-- acknowledgement is returned, indicating that the message was
-- received by the receiver
-- the Kernel possibly changes the process priority
--
POSTCONDITIONS:
-- the message is sent to the intended receiver
-- the priority of this process is its resumption priority
-- the process is unblocked and is either suspended or running depending
-- on the discretion of the Kernel Scheduler
--
ERROR PROCESSING:
-- when send_message_and_wait is called within an interrupt handler:
-- the Kernel raises illegal_context_for_call
--
POSTCONDITIONS:
-- the call is rejected; no message is sent; no change in priority
-- is registered
--
-- when this primitive is called with an undeclared process ID
-- the Kernel raises illegal_process_identifier_exception
--
POSTCONDITIONS:
-- the call is rejected; no message is sent; no change in priority
-- is registered
--
-- when the specified receiver of the message does not actually receive
-- the message (e.g., the message is dropped or the process dies):
-- the Kernel raises message_not_received_exception
--
POSTCONDITIONS:

```

```

--/ the sent message is never received by the receiver
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when the remote timeout expires:
--/ the Kernel raises message_limited_out exception
--/ POSTCONDITIONS:
--/ the sent message is never received by the receiver
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when a network problem is detected:
--/ the Kernel raises network_failure exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when the local Kernel detects that the specified receiver is dead:
--/ the Kernel raises receiver_dead exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when a process specifies itself as the intended receiver:
--/ the Kernel raises receiver_is_sender exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ when the local Kernel has no knowledge of the specified receiver:
--/ the Kernel raises receiver_never_existed exception
--/ POSTCONDITIONS:
--/ the send does not occur
--/ the priority of this process is its resumption priority
--/ the process is unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/ pragma (page);

```

```

procedure Send_Message_And_Wait (
Receiver : In Pib.Process_Identifier;
Message_Tag : In Cg.Message_Tag_Type;
Message_Length : In Cg.Message_Length_Type := 0;
Message_Text : In Hw.Address := Null_Hw_Address;
Timeout_At : In Tg.Epoch_Time;
Resumption_Priority : In St.Priority := St.Current_Process_Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ send_message_and_wait
--/
--/ DESCRIPTION:
--/ does a synchronous send of a message to a specified receiver; the
--/ calling process blocks until a Kernel-to-Kernel acknowledgement of
--/ message receipt (or lack thereof) by the receiving process is
--/ returned or until the specific Yepoch (absolute) time has arrived
--/
--/ --REQ: 7.1.21; 7.1.22; 7.1.25; 7.1.26
--/ --REQ: 9.1.6; 9.1.8; 9.1.22; 10.1.2
--/ --REQ: 10.1.3; 10.1.4; 10.1.5; 10.1.6; 10.1.8; 10.1.9; 10.1.10; 10.1.11
--/ --REQ: 10.1.25; 10.1.27; 10.1.32
--/ --REQ: 11.1.12; 14.1.5
--/
--/ --PRIM: 10.1.31; 19.1.2
--/
--/ --TIME: 10.2.3; 10.2.4; 10.2.7; 10.2.8; 10.2.11
--/ --TIME: 14.2.3; 14.2.5; 14.2.6
--/
--/ PARAMETERS:
--/ receiver
--/ the process ID of the intended receiver of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing message-specific data
--/
--/ default value:
--/ 0
--/
--/ message_text
--/ the address at which the message-specific data begin; the Kernel
--/ collects message_length bytes of data beginning at the address
--/ specified by message_text
--/
--/ default value:
--/

```

```

../ null_hw_address
../
../ timeout at
../ absolute time at which the attempt to send is abandoned and the
../ process becomes unblocked
../
../ default value:
../ none
../
../ resumption_priority
../ the priority to be assigned to this process when the
../ acknowledgement is received (i.e., the process becomes unblocked)
../
../ default value:
../ the calling process's current priority
../
../ PRECONDITIONS:
../ invoked from a non-blocking context
../ the receiver is not the sender
../
../ ACTIONS:
../ the Kernel validates the receiver field
../ the Kernel indicates that a Kernel-to-Kernel acknowledgement is
../ required when the receiver receives the message
../ the Kernel bundles the elapsed time until the specified epoch_time
../ value (i.e., the local Kernel converts "until epoch_time" to the
../ appropriate elapsed time duration) with the message (i.e., the
../ timeout is a timeout at the receiver's site and measured with
../ respect to the remote processor's clock)
../ the Kernel sends the message from the caller to the specified
../ receiver using local optimization if possible
../ the calling process blocks until a Kernel-to-Kernel positive
../ acknowledgement is returned, indicating that the message was
../ received by the receiver
../ the Kernel possibly changes the process priority
../
../ POSTCONDITIONS:
../ the message is sent to the intended receiver
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running depending
../ on the discretion of the Kernel Scheduler
../
../ ERROR PROCESSING:
../ when send_message and wait is called within an interrupt handler:
../ the Kernel raises illegal_context_for_call
../
../ POSTCONDITIONS:
../ the call is rejected; no message is sent; no change in priority
../ is registered
../ when this primitive is called with an undeclared process ID
../ the Kernel raises illegal_process_identifier_exception
../
../ POSTCONDITIONS:
../ the call is rejected; no message is sent; no change in priority
../ is registered
../ when the specified receiver of the message does not actually receive
../ the message (e.g., the message is dropped or the process dies):

```

```

../ the Kernel raises message_not_received_exception
../
../ POSTCONDITIONS:
../ the sent message is never received by the receiver
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler
../ when the remote timeout expires:
../ the Kernel raises message_timed_out_exception
../
../ POSTCONDITIONS:
../ the sent message is never received by the receiver
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler
../ when a network problem is detected:
../ the Kernel raises network_failure_exception
../
../ POSTCONDITIONS:
../ the send does not occur
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler
../ when the local Kernel detects that the specified receiver is dead:
../ the Kernel raises receiver_dead_exception
../
../ POSTCONDITIONS:
../ the send does not occur
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler
../ when a process specifies itself as the intended receiver:
../ the Kernel raises receiver_is_sender_exception
../
../ POSTCONDITIONS:
../ the send does not occur
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler
../ when the local Kernel has no knowledge of the specified receiver:
../ the Kernel raises receiver_never_existed_exception
../
../ POSTCONDITIONS:
../ the send does not occur
../ the priority of this process is its resumption priority
../ the process is unblocked and is either suspended or running
../ depending on the discretion of the Kernel Scheduler

```

```

../ pragma (page);

```

```

procedure Receive_Message (
  Sender : out Pib.Process_Identifier;
  Message_Tag : out Cg.Message_Tag_Type;
  Message_Length : out Cg.Message_Length_Type;
  Message_Buffer : in Hw_Address;
  Buffer_Size : in Cg.Message_Length_Type;
  Resumption_Priority : in St.Priority := St.Current_Process_Priority;
  Messages_Lost : out Boolean);
--/
--/ SUBPROGRAM NAME:
--/ receive_message
--/
--/ DESCRIPTION:
--/ does a receive of a message from its message queue; the calling
--/ process blocks until a message is available for receipt
--/
--/ --REQ: 9.1.6; 9.1.8; 9.1.22
--/ --REQ: 10.1.12; 10.1.13; 10.1.14; 10.1.15; 10.1.20
--/ --REQ: 10.1.21; 10.1.24; 10.1.26; 10.1.35; 10.1.36
--/ --REQ: 11.1.12; 14.1.5
--/
--/ --PRIM: 19.1.3
--/
--/ --TIME: 10.2.9; 10.2.10
--/ --TIME: 14.2.3; 14.2.5; 14.2.6
--/
--/ PARAMETERS:
--/ sender
--/ the process ID of the sender of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing r-message-specific data
--/
--/ default value:
--/ none
--/
--/ message_buffer
--/ the address at which the message-specific data are to be put;
--/ the Kernel places no more than Buffer_Size bytes of data beginning
--/ at the address specified by message_buffer
--/
--/ default value:
--/ none
--/

```

```

--/ buffer_size
--/ the number of bytes of text into which message-specific data are
--/ placed
--/
--/ default value:
--/ none
--/
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked
--/
--/ default value:
--/ the calling process's current priority
--/
--/ messages_lost
--/ an indicator of whether or not the incoming message queue is
--/ sufficiently full that newly arriving messages were unable to fit
--/ into it in a non-destructive manner before the current receive
--/ values include:
--/ true (messages have been lost)
--/ false (messages have not been lost)
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ if a message is available, then the process doesn't block; otherwise,
--/ the calling process blocks (if permitted, see ERROR PROCESSING
--/ below) until a message is available;
--/ the Kernel ensures that the buffer_size is sufficient to hold the
--/ message text consisting of message_length bytes
--/ the Kernel sends the appropriate Kernel-to-Kernel acknowledgement of
--/ message receipt (if required)
--/ the Kernel places the message text into the message_buffer and
--/ sets the sender, message_tag, and the message_length
--/ the Kernel sets the value of messages_lost appropriately
--/ the Kernel possibly changes the process priority
--/
--/ POSTCONDITIONS:
--/ the calling process has the message text and message attributes
--/ the calling process has an indication of whether a previous message
--/ was lost owing to the incoming message queue becoming full
--/ the message is removed from the receiver's incoming message queue
--/ the Kernel-to-Kernel acknowledgement has been sent to the message
--/ sender if required
--/ the priority of this process is its resumption priority
--/ if a message was available, the process is either suspended or
--/ running depending on the discretion of the Kernel Scheduler;
--/ otherwise (if the process had been blocked waiting for a message)
--/ the process becomes unblocked and is either suspended or running
--/ depending on the discretion of the Kernel Scheduler
--/
--/ ERROR PROCESSING:

```

```

--/ when the buffer_size parameter is too small (i.e., it is smaller than
--/ the message length);
--/ the Kernel raises buffer_too_small_for_message_exception
--/ POSTCONDITIONS:
--/ the message "envelope" (i.e., its attributes: sender, message
--/ tag, message length) is returned, no message text is
--/ returned
--/ the calling process has an indication of whether a previous
--/ message was lost owing to the incoming message queue
--/ becoming full
--/ the message is removed from the receiver's incoming message
--/ queue
--/ the Kernel-to-Kernel negative acknowledgement has been sent to
--/ the message sender if required
--/ the priority of this process is its resumption priority
--/ If a message was available, the process is either suspended or
--/ running depending on the discretion of the Kernel Scheduler;
--/ otherwise (if the process had been blocked waiting for a
--/ message) the process becomes unblocked and is either
--/ suspended or running depending on the discretion of the
--/ Kernel Scheduler
--/ when receive_message is called within an interrupt handler:
--/ the Kernel raises illegal_context_for_call
--/ POSTCONDITIONS:
--/ the call is rejected; no message is received; no change in
--/ priority is registered
--/
--/ NOTES:
--/ When a receive fails for any reason (expiry of time out,
--/ message too big, illegal context for call, etc.) the
--/ following values will be set in the output parameters:
--/ sender := PTB.null_process
--/ message_length := CG.message_length_type(0)
--/ message_tag := CG.message_tag_type(0)
--/ messages_lost := false
--/ Since the various exceptions can be disabled, this is
--/ another way of testing for a failed receive.

```

```

--/ pragma (page);

```

```

procedure Receive_Message (
  Sender : out Ptb.Process_Identifier;
  Message_Tag : out Cg.Message_Tag_Type;
  Message_Length : out Cg.Message_Length_Type;
  Message_Buffer : in Hw.Address;
  Buffer_Size : in Cg.Message_Length_Type;
  Resumption_Priority : in St.Priority := St.Current_Process_Priority;
  Timeout_After : in Tg.Elapsed_Time;
  Messages_Lost : out Boolean);
--/ =====
--/ SUBPROGRAM NAME:
--/ receive_message
--/
--/ DESCRIPTION:
--/ does a receive of a message from its message queue; the calling
--/ process blocks until a message is available for receipt or until the
--/ specified elapsed time has elapsed
--/
--/ --REQ: 9.1.6; 9.1.8; 9.1.22
--/ --REQ: 10.1.12; 10.1.13; 10.1.14; 10.1.15; 10.1.16; 10.1.18
--/ --REQ: 10.1.20; 10.1.21; 10.1.24; 10.1.26; 10.1.35; 10.1.36
--/ --REQ: 11.1.12; 14.1.5
--/
--/ --PRIM: 19.1.3
--/
--/ --TIME: 10.2.9; 10.2.10
--/ --TIME: 14.2.3; 14.2.5; 14.2.6
--/
--/ PARAMETERS:
--/ sender
--/ the process ID of the sender of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing message-specific data
--/
--/ default value:
--/ none
--/
--/ message_buffer
--/ the address at which the message-specific data are to be put;
--/ the Kernel places no more than buffer_size bytes of data beginning
--/ at the address specified by message_buffer
--/
--/ default value:

```

```

../ none
../ buffer size
../ the number of bytes of text into which message-specific data are
  placed
../ default value:
../ none
../ timeout after
  duration after which the process resinds its attempt to receive
  a message and becomes unblocked
../ default value:
../ none
../ resumption priority
  the priority to be assigned to this process when it becomes
  unblocked
../ default value:
../ the calling process's current priority
../ messages lost
  an indication of whether or not the incoming message queue is
  sufficiently full that newly arriving messages were unable to fit
  into it in a non-destructive manner before the current receive
  values include:
  true (messages have been lost)
  false (messages have not been lost)
../ PRECONDITIONS:
../ NONE
../ ACTIONS:
../ If a message is available, then the process doesn't block, otherwise,
  the calling process blocks (if permitted, see ERROR PROCESSING
  below) until a message is available or until the specified timeout
  expires (for the latter case, see ERROR PROCESSING below)
../ If a message is not available and the timeout is for zero or less
  elapsed time, then the process does not wait for a message (see
  ERROR PROCESSING below)
../ when a message is available within the specified elapsed time:
  the Kernel ensures that the buffer size is sufficient to hold the
  message text consisting of message_length bytes
../ the Kernel sends the appropriate Kernel-to-Kernel acknowledgement of
  message receipt (if required)
../ the Kernel places the message text into the message_buffer and
  sets the sender_message_tag, and the message_length
../ the Kernel sets the value of messages_lost appropriately
../ the Kernel possibly changes the process priority
../ POSTCONDITIONS:
../ the calling process has the message text and message attributes

```

```

../ the calling process has an indication of whether a previous message
  was lost owing to the incoming message queue becoming full
../ the message is removed from the receiver's incoming message queue
  the Kernel-to-Kernel acknowledgement has been sent to the message
  sender
../ the priority of this process is its resumption priority
  If a message was available, the process is either suspended or
  running depending on the discretion of the Kernel Scheduler;
  otherwise (if the process had been blocked waiting for a message)
  the process becomes unblocked and is either suspended or running
  depending on the discretion of the Kernel Scheduler
../ ERROR PROCESSING:
../ when the buffer_size parameter is too small (i.e., it is smaller than
  the message_length):
  the Kernel raises buffer_too_small_for_message_exception
../ POSTCONDITIONS:
  the message "envelope" (i.e., its attributes: sender, message
  tag, message_length) is returned, no message text is
  returned
../ the calling process has an indication of whether a previous
  message was lost owing to the incoming message queue
  becoming full
../ the message is removed from the receiver's incoming message
  queue
../ the Kernel-to-Kernel negative acknowledgement has been sent to
  the message sender if required
../ the priority of this process is its resumption priority
  If a message was available, the process is either suspended or
  running depending on the discretion of the Kernel Scheduler;
  otherwise (if the process had been blocked waiting for a
  message) the process becomes unblocked and is either
  suspended or running depending on the discretion of the
  Kernel Scheduler
../ when receive_message is called within an interrupt handler:
  the Kernel raises illegal_context_for_call
../ POSTCONDITIONS:
  the call is rejected; no message is received; no change in
  priority is registered
  when the receive_timeout expires:
  the Kernel raises message_timed_out_exception
../ POSTCONDITIONS:
  no receive is done (there is nothing to receive within the
  specified elapsed time)
  the priority of this process is its resumption priority
  when the timeout value is less than or equal to zero and there is no
  message in the receiver's incoming message queue:
  the Kernel raises no_message_available_exception
../ POSTCONDITIONS:
  no receive is done (there is nothing to receive within the
  specified elapsed time)
  the priority of this process is its resumption priority
../ NOTES:
  When a receive fails for any reason (expiry of time out,

```



```

--/ message too big, illegal context for call, etc.) the
--/ following values will be set in the output parameters:
--/ sender := PTB.null_process
--/ message_length := CG.message_length_type(0)
--/ message_tag := CG.message_tag_type(0)
--/ messages_lost := false
--/ Since the various exceptions can be disabled, this is
--/ another way of testing for a failed receive.
--/
--/ pragma (page);

```

```

procedure Receive_Message (
  Sender : out Ptb.Process_Identifier;
  Message_Tag : out Cg.Message_Tag_Type;
  Message_Length : out Cg.Message_Length_Type;
  Message_Buffer : in Hw.Address;
  Buffer_Size : in Cg.Message_Length_Type;
  Timeout_At : in Tg.Epoch_Time;
  Resumption_Priority : in St.Priority := St.Current_Process_Priority;
  Messages_Lost : out Boolean);
--/
--/ SUBPROGRAM NAME:
--/ receive_message
--/
--/ DESCRIPTION:
--/ does a receive of a message from its message queue; the calling
--/ process blocks until a message is available for receipt or until the
--/ specified elapsed time has elapsed
--/
--/ --REQ: 9.1.6; 9.1.8; 9.1.22
--/ --REQ: 10.1.12; 10.1.13; 10.1.14; 10.1.15; 10.1.17; 10.1.18
--/ --REQ: 10.1.20; 10.1.24; 10.1.26; 10.1.35; 10.1.38
--/ --REQ: 11.1.12; 14.1.5
--/
--/ --PRIM: 19.1.3
--/
--/ --TIME: 10.2.9; 10.2.10
--/ --TIME: 14.2.3; 14.2.5; 14.2.6
--/
--/ PARAMETERS:
--/ sender
--/ the process ID of the sender of this message
--/
--/ default value:
--/ none
--/
--/ message_tag
--/ an application-defined tag that allows the receiver to interpret
--/ the message in the format the sender intended
--/
--/ default value:
--/ none
--/
--/ message_length
--/ the number of bytes of text that should be interpreted as
--/ containing message-specific data
--/
--/ default value:
--/ none
--/
--/ message_buffer
--/ the address at which the message-specific data are to be put;
--/ the kernel places no more than buffer_size bytes of data beginning
--/ at the address specified by message_buffer
--/
--/ default value:

```

```

--/ none
--/
--/ buffer_size
--/ the number of bytes of text into which message-specific data are
--/ placed
--/
--/ default_value:
--/ none
--/
--/ timeout_at
--/ absolute time at which the process rescinds its attempt to receive
--/ a message and becomes unblocked
--/
--/ default_value:
--/ none
--/
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked
--/
--/ default_value:
--/ the calling process's current priority
--/
--/ messages_lost
--/ an indication of whether or not the incoming message queue is
--/ sufficiently full that newly arriving messages were unable to fit
--/ into it in a non-destructive manner before the current receive
--/ values include:
--/ true (messages have been lost)
--/ false (messages have not been lost)
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ if a message is available, then the process doesn't block; otherwise,
--/ the calling process blocks (if permitted, see ERROR PROCESSING
--/ below) until a message is available or until the specified timeout
--/ expires (for the latter case, see ERROR PROCESSING below)
--/ if a message is not available and the timeout is for a non-future
--/ epoch time, then the process does not wait for a message (see
--/ ERROR PROCESSING below)
--/ when a message is available by the specified epoch time:
--/ the kernel ensures that the buffer_size is sufficient to hold the
--/ message text consisting of message_length bytes
--/ the kernel sends the appropriate kernel-to-kernel acknowledgement of
--/ message receipt (if required)
--/ the kernel places the message text into the message_buffer and
--/ sets the sender, message_tag, and the message_length
--/ the kernel sets the value of messages_lost appropriately
--/ the kernel possibly changes the process priority
--/
--/ POSTCONDITIONS:
--/ the calling process has the message text and message attributes

```

```

--/ the calling process has an indication of whether a previous message
--/ was lost owing to the incoming message queue becoming full
--/ the message is removed from the receiver's incoming message queue
--/ the kernel-to-kernel acknowledgement has been sent to the message
--/ sender
--/ the priority of this process is its resumption priority
--/ if a message was available, the process is either suspended or
--/ running depending on the discretion of the kernel scheduler;
--/ otherwise (if the process had been blocked waiting for a message)
--/ the process becomes unblocked and is either suspended or running
--/ depending on the discretion of the kernel scheduler
--/
--/ ERROR PROCESSING:
--/ when the buffer_size parameter is too small (i.e., it is smaller than
--/ the message_length):
--/ the kernel raises buffer_too_small_for_message_exception
--/
--/ POSTCONDITIONS:
--/ the message "envelope" (i.e., its attributes: sender, message
--/ tag, message_length) is returned, no message text is
--/ returned
--/ the calling process has an indication of whether a previous
--/ message was lost owing to the incoming message queue
--/ becoming full
--/ the message is removed from the receiver's incoming message
--/ queue
--/ the kernel-to-kernel negative acknowledgement has been sent to
--/ the message sender if required
--/ the priority of this process is its resumption priority
--/ if a message was available, the process is either suspended or
--/ running depending on the discretion of the kernel scheduler;
--/ otherwise (if the process had been blocked waiting for a
--/ message) the process becomes unblocked and is either
--/ suspended or running depending on the discretion of the
--/ kernel scheduler
--/ when receive_message is called within an interrupt handler:
--/ the kernel raises illegal_context_for_call
--/
--/ POSTCONDITIONS:
--/ the call is rejected; no message is received; no change in
--/ priority is registered
--/ when the receive_timeout expires:
--/ the kernel raises message_timed_out_exception
--/
--/ POSTCONDITIONS:
--/ no receive is done (there is nothing to receive within the
--/ specified elapsed time)
--/ the priority of this process is its resumption priority
--/ when the timeout value is for a non-future epoch time and there is no
--/ message in the receiver's incoming message queue:
--/ the kernel raises no_message_available_exception
--/
--/ POSTCONDITIONS:
--/ no receive is done (there is nothing to receive before the
--/ specified epoch time)
--/ the priority of this process is its resumption priority
--/
--/ NOTES:
--/ When a receive fails for any reason (expiry of time out,

```

```

--/ message too big. illegal context for call, etc.) the
--/ following values will be set in the output parameters:
--/ sender := PTB null process
--/ message_length := CG.message_length_type(0)
--/ message_tag := CG.message_tag_type(0)
--/ messages_lost := false
--/ Since the various exceptions can be disabled, this is
--/ another way of testing for a failed receive.
--/
--/ pragma (page);

```

```

procedure Allocate_Device_Receiver (
  Receiver_Process_ID : in Pto.Process_Identifier;
  Device_ID : in Ng.Processor_Identifier);
--/ =====
--/ SUBPROGRAM NAME:
--/ allocate_device_receiver
--/
--/ DESCRIPTION:
--/ assigns a specified process to be the receiver of all messages from
--/ a specified non-Kernel device
--/ --REQ: 7.1.14
--/ --PRIM: 10.1.31; 16.1.4
--/ --TIME: 7.2.4
--/
--/ PARAMETERS:
--/ receiver_process_ID
--/ the process ID of the process to receive messages from the
--/ non-Kernel device
--/ default value:
--/ none
--/
--/ device_ID
--/ the processor ID of the device from which messages are sent
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ receiver_process_ID was generated by a previous call to
--/ declare_process
--/ the device ID corresponds to an existing entry in the NCT for that
--/ device (again, the 1-to-1 mapping of non-Kernel devices to
--/ non-Kernel processes)
--/
--/ ACTIONS:
--/ the Kernel registers the specified receiver_process_ID as the sole
--/ receiver of messages from the specified device
--/
--/ POSTCONDITIONS:
--/ update to the Kernel's routing tables for messages from non-Kernel
--/ devices
--/
--/ ERROR PROCESSING:
--/ when the Kernel does not recognize the device ID as a non-Kernel
--/ device:
--/ the Kernel raises no such_device_exists_exception
--/ POSTCONDITIONS:
--/ the (re)binding does not occur
--/ when this primitive is called with an undeclared process ID
--/ the Kernel raises illegal_process_identifier_exception
--/ POSTCONDITIONS:

```

```

--/ the (re)binding does not occur
--/ when any call other than the first call is made:
--/ the Kernel raises replacing_previous_allocation_exception
--/ POSTCONDITIONS:
--/ the rebinding occurs
--/ the Kernel routing tables are updated accordingly
--/
--/ =====
end Generic_Communication_Management;
pragma page;

```

```

--/ .....
--/ MODULE NAME:
--/ communication_management
--/
--/ MODULE TYPE:
--/ package specification
--/
--/ MODULE PURPOSE:
--/ an example of (and a template for) an instantiation of the generic
--/ package generic_communication_management
--/
--/ -----
--/ MODULE DESCRIPTION:
--/ package instantiating the capabilities to send and receive messages
--/ among processes where:
--/ error checking for buffer too small for message is enabled
--/ error checking for illegal context for call is enabled
--/ error checking for illegal process identifier is enabled
--/ error checking for message not received is enabled
--/ error checking for message timed out is enabled
--/ error checking for network failure is enabled
--/ error checking for no message available is enabled
--/ error checking for no such device exists is enabled
--/ error checking for receiver dead is enabled
--/ error checking for receiver never existed is enabled
--/ error checking for replacing_previous_allocation is enabled
--/
--/ MODULE CONTENTS:
--/ communication_management (package specification)
--/ allocate_device_receiver (procedure)
--/ buffer_too_small_for_message (exception)
--/ illegal_context_for_call (exception)
--/ illegal_process_identifier (exception)
--/ message_not_received (exception)
--/ message_timed_out (exception)
--/ network_failure (exception)
--/ no_message_available (exception)
--/ no_such_device_exists (exception)
--/ receive_message (overloaded (3) procedure)
--/ receiver_dead (exception)
--/ receiver_is_sender (exception)
--/ receiver_never_existed (exception)
--/ replacing_previous_allocation (exception)
--/ send_message (procedure)
--/ send_message_and_wait (overloaded (3) procedure)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ none
--/
--/ USER'S MANUAL:
--/ Kernel User's Manual
--/
--/ TESTING AND VALIDATION:

```

```

package Communication_Management

is new Generic Communication_Management (
  Buffer Too Small For Message_Enabled => True,
  Illegal Context For Call_Enabled => True,
  Illegal Process Identifier_Enabled => True,
  Message Not Received_Enabled => True,
  Message Timed Out_Enabled => True,
  Network Failure_Enabled => True,
  No Message Available_Enabled => True,
  No Such Device Exists_Enabled => True,
  Receiver Dead_Enabled => True,
  Receiver Is Sender_Enabled => True,
  Receiver Never Existed_Enabled => True,
  Replacing Previous_Allocation_Enabled => True);
pragma (page);

```

```

--/ none
--/
--/ NOTES:
--/ none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/ 18aug88 bamborg created (compiles clean)
--/ 29jun89 dkt Bug 0040_SEI: Added check for illegal PIDs
--/
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
--/
with Generic Communication_Management;
pragma Elaborate (Generic Communication_Management);
.. to get see MODULE CONTENTS
--/ .....
--/ package short name CM
--/ .....

```

```

./ .....
./ MODULE NAME:
./   generic_process_attribute_modifiers
./
./ MODULE TYPE:
./   generic_package_specification
./
./ MODULE PURPOSE:
./   support the modification of attributes of already existing processes
./
./ .....
./ MODULE DESCRIPTION:
./   collection of subprograms to modify the attributes of already existing
./   processes
./
./ MODULE CONTENTS:
./   generic_process_attribute_modifiers (generic package specification)
./   die (procedure)
./   illegal_context_for_call (exception)
./   illegal_context_for_call_enabled (generic formal parameter)
./   illegal_process_identifier (exception)
./   illegal_process_identifier_enabled (generic formal parameter)
./   kill (procedure)
./   set_process_preemption (procedure)
./   set_process_priority (procedure)
./   wait (overloaded (2) procedure)
./
./ REFERENCES:
./   DESIGN DOCUMENTS:
./   Kernel Facilities Definition
./   --REQ: 5.1.8, 6.1.9, 7.1.1, 7.1.2, 7.1.13
./   --PRIM: 10.1.31, 16.1.5, 18.1.1, 18.1.3, 18.1.5
./   --TIME: N/A
./
./ USER'S MANUAL:
./   Kernel User's Manual
./
./ TESTING AND VALIDATION:
./   Unit testing
./
./ NOTES:
./   In die, set_process_preemption, set_process_priority, and wait:
./   If checking for illegal_context_for_call is disabled, the application
./   is asserting that it guarantees that the call is NOT from within an
./   interrupt handler; if the call IS from within an interrupt handler
./   and checking is disabled, this procedure simply returns (the
./   interrupt handler does NOT die)
./   In kill:
./   If checking for illegal_process_identifier is disabled, the application
./   will not be notified if an attempt is made to kill a non-existent
./   process (and that an attempt to do so will be a null action).
./
./ .....
./ MODIFICATION HISTORY:
./   18aug88 bamberg created (complies clean)

```

```

./ 24sep88 ova changed to use package short names (and conform
./ with package body )
./ 15feb89 bamberg update documentation per bug 0012, SEI
./ 29jun89 ova Bug 0040, SEI: add checks for illegal PIDs
./ 27aug89 bamberg final doc; check for 01sep89 freeze
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./
./ The following copyright must be included in this software and
./ all software utilizing this software.
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy or functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....

```

pragma (page);

```

with Kernel_Exceptions;
.. to get: illegal_context_for_call
.. illegal_process_identifier

with Process_Table;
.. to get: process_identifier
.. pending_activity_type

with Schedule.Types;
.. to get: current_process_priority
.. preemption
.. priority

with Time_Globals;
.. to get: elapsed_time
.. epoch_time
.. to kernel_time

generic
..
.. ERROR checking conditional compilation flags
..
.. REQ: 5.1.8
illegal_context_for_call_enabled : in Boolean := True;
illegal_process_identifier_enabled : in Boolean := True;
..
.. package short name: GPAM
..
package Generic_Process_Attribute_Modifiers
is
package Ke
renames Kernel_Exceptions;
package Ptb
renames Process_Table;
package St
renames Schedule.Types;
package Tg

```

```

renames Time_Globals;
..REQ: 6.1.8; 7.1.1; 7.1.2; 7.1.13

.. illegal_context_for_call_raised_by:
.. de
.. set_process_preemption
.. set_process_priority
.. wait
..PRIM: 16.1.5; 18.1.1; 18.1.3; 18.1.5
illegal_context_for_call_exception
renames Ke.illegal_context_for_call;

..
.. illegal_process_identifier_raised_by:
.. kill
..PRIM: 10.1.31
illegal_process_identifier_exception
renames Ke.illegal_process_identifier;

pragma (page);

```

```

procedure Die;
--/ =====
--/ SUBPROGRAM NAME:
--/ die
--/
--/ DESCRIPTION:
--/ terminate the calling process
--/
--/ --REQ: 7.1.18, 7.1.21, 7.1.25, 7.1.26, 8.1.12
--/
--/ --PRIM: 16.1.5
--/
--/ --TIME: 7.2.2
--/
--/ PARAMETERS:
--/ none
--/
--/ PRECONDITIONS:
--/ none (other than the calling process must not already be dead, but
--/ if it were dead, it couldn't make this call)
--/
--/ ACTIONS:
--/ the Kernel updates this process's state to dead (i.e., it may never
--/ be rescheduled)
--/ no additional messages are accepted for the dying process
--/ all pending messages for this process are discarded
--/ all pending messages for this process that require a 1 acknowledgement
--/ are negatively acknowledged
--/ any binding of this process to any non-Kernel device is unbound
--/
--/ POSTCONDITIONS:
--/ this process's state is dead
--/
--/ ERROR PROCESSING:
--/ when die is called within an interrupt handler:
--/ the Kernel raises illegal_context_for_call
--/
--/ POSTCONDITIONS:
--/ the call is rejected; nothing dies
--/
--/ EXAMPLES:
--/ <sample calls and explanations>
--/
--/ =====
pragma (page);

```

```

procedure Kill (
  Process_Id : in Pib.Process_Identifier);
--/ =====
--/ SUBPROGRAM NAME:
--/ Kill
--/
--/ DESCRIPTION:
--/ asynchronously aborts the specified Kernel process
--/
--/ --REQ: 7.1.18, 7.1.19, 7.1.20, 7.1.22, 7.1.25, 7.1.26, 8.1.12
--/
--/ --REQ: 14.1.5
--/
--/ --PRIM: 10.1.31, 16.1.6
--/
--/ --TIME: 7.2.2, (7.2.3)
--/
--/ --TIME: 14.2.3, 14.2.5, 14.2.6
--/
--/ PARAMETERS:
--/ process ID
--/ the ID of the process to be killed (may be local or remote)
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ the Kernel updates the specified process's state to dead
--/ (i.e., it may never be rescheduled) (this is done asynchronously
--/ if the process to be killed is not a local process)
--/ the appropriate Kernel data structures are updated to note that the
--/ specified process is dead IF AND ONLY IF the process is local
--/ no additional messages are accepted for the process being killed
--/ all pending messages for the specified process are discarded
--/ all pending messages for the specified process that require an
--/ acknowledgement are negatively acknowledged
--/ any binding of the specified process to any non-Kernel device is
--/ unbound
--/ If the process is already dead, nothing happens
--/
--/ POSTCONDITIONS:
--/ the specified process's state is dead
--/
--/ =====
--/ ERROR PROCESSING:
--/ If an illegal process identifier is passed as a parameter
--/ the exception illegal_process_identifier_is_raised (if enabled)
--/ no action is performed (if not)
--/
--/ =====
pragma (page);

```



```

procedure Set_Process_Preemption (
  Preemptible : In St_Preemption := St_Enabled);
-- SUBPROGRAM NAME:
-- set_process_preemption
--
-- DESCRIPTION:
-- changes the preemption status of the calling process
--
-- REQ: 9.1.12; 9.1.22
--
-- PRIM: 18.1.1
--
-- TIME: 9.2.2
--
-- PARAMETERS:
-- preemptible
-- Indication of whether or not a process may be preempted by
-- another process of the same priority
--
-- values include:
-- enabled (the process is preemptible)
-- disabled (the process is not preemptible)
--
-- default value:
-- enabled (the process is preemptible)
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- If the preemption is not changed, nothing happens
-- If preemption is being disabled, then time slicing is disregarded for
-- this process, and this process continues execution
-- If preemption is being enabled, the Kernel changes the process state
-- to suspended, forcing a reschedule point
-- the Kernel records the specified value for this process's preemption
-- status
--
-- POSTCONDITIONS:
-- the process's preemption status is as specified in the call
-- the process state is either suspended or running, depending on the
-- discretion of the Kernel Scheduler
--
-- ERROR PROCESSING:
-- when this primitive is called in an interrupt handler,
-- the Kernel raises illegal_context_for_call_exception
--
-- POSTCONDITIONS:
-- the call is rejected; the set does not occur
--
pragma (page);

```

```

procedure Set_Process_Priority (
  New_Priority : In St_Priority);
-- SUBPROGRAM NAME:
-- set_process_priority
--
-- DESCRIPTION:
-- changes the priority of the calling process
--
-- REQ: 9.1.6; 9.1.8; 9.1.22
--
-- PRIM: 18.1.3
--
-- TIME: 9.2.1
--
-- PARAMETERS:
-- new_priority
-- priority at which the process is now to run
--
-- default value:
-- none
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- If there is no change in priority, this simply returns
-- If the priority is raised,
-- the Kernel just records the specified value
-- if the process priority is lowered, the Kernel changes the process
-- state to suspended and records the specified value for the
-- process's priority
--
-- POSTCONDITIONS:
-- the process's priority is as specified in the call
-- the process state is either suspended or running depending on the
-- discretion of the Kernel Scheduler
--
-- ERROR PROCESSING:
-- when this primitive is called in an interrupt handler,
-- the Kernel raises illegal_context_for_call_exception
--
-- POSTCONDITIONS:
-- the call is rejected; the set does not occur
--
pragma (page);

```

```

--/ the Kernel raises illegal_context for call exception
--/ POSTCONDITIONS:
--/ the wait does not occur
--/ no change in priority is registered
--/
--/ =====
--/ pragma (page);

```

```

procedure Wait (
  Until Epoch Time : In Tg.Epoch Time;
  Resumption Priority : In St.Priority := St.Current.Process.Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ wait
--/
--/ DESCRIPTION:
--/ allows the caller to suspend its own execution
--/
--/ --REQ: 9.1.6; 9.1.8; 9.1.14; 9.1.16; 9.1.17; 9.1.22; 9.1.29; 11.1.12
--/ --PRIM: 18.1.5
--/ --TIME: 9.2.3
--/
--/ PARAMETERS:
--/ until epoch time
--/ absolute time at which the process becomes unblocked
--/
--/ default value:
--/ none
--/
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked at until_epoch_time
--/
--/ default value:
--/ the calling process's current priority
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ If the epoch time until which to wait is now or in the past, the Kernel
--/ changes the process state of the caller to suspended, forcing a
--/ reschedule
--/
--/ If the epoch time until which to wait is in the future,
--/ the Kernel changes the process state of the caller to blocked
--/ the Kernel notes that the process is to be unblocked at the specified
--/ epoch time
--/
--/ the Kernel possibly changes the process priority
--/
--/ POSTCONDITIONS:
--/ the calling process is suspended or blocked, depending on the value of
--/ the epoch time until which to wait
--/
--/ the epoch time at which the process is to be unblocked is noted by
--/ the Kernel
--/
--/ the resumption priority at which the process is to be unblocked is
--/ noted by the Kernel
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in a context that prohibits blocking
--/ (e.g., within an interrupt handler) and this call would block
--/ (i.e., the epoch time is for some future time);

```

```

procedure Wait (
  For_Elapsed_Time : In Tg.Elapsed_Time;
  Resumption_Priority : In St.Priority := St.Current_Process.Priority);
--/
--/ SUBPROGRAM NAME:
--/ wait
--/
--/ DESCRIPTION:
--/ allows the caller to suspend its own execution
--/
--/ REQ: 9.1.6, 9.1.8, 9.1.14, 9.1.15, 9.1.17, 9.1.22, 9.1.29, 11.1.12
--/
--/ PRIM: 18.1.5
--/
--/ TIME: 9.2.3
--/
--/ PARAMETERS:
--/ for elapsed time
--/ duration after which the process becomes unblocked
--/
--/ default value:
--/ none
--/
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked after for_elapsed_time
--/
--/ default value:
--/ the calling process's current priority
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ If the elapsed time for which to wait is now or in the past, the Kernel
--/ changes the process state of the caller to suspended, forcing a
--/ reschedule
--/
--/ If the elapsed time for which to wait is in the future:
--/ the Kernel changes the process state of the caller to blocked
--/ the Kernel notes that the process is to be unblocked at the specified
--/ elapsed time
--/
--/ the Kernel possibly changes the process priority
--/
--/ POSTCONDITIONS:
--/ the calling process is suspended or blocked, depending on the value of
--/ the elapsed time until which to wait
--/
--/ the elapsed time at which the process is to be unblocked is noted by
--/ the Kernel
--/
--/ the resumption priority at which the process is to be unblocked is
--/ noted by the Kernel
--/
--/ POSTCONDITIONS:
--/ the calling process is blocked
--/
--/ the elapsed time after which the process is to be unblocked is noted
--/ by the Kernel
--/
--/
--/ the resumption priority at which the process is to be unblocked is
--/ noted by the Kernel
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in a context that prohibits blocking
--/ (e.g., within an interrupt handler) and this call would block
--/ (i.e., the elapsed time is greater than zero):
--/ the Kernel raises Illegal_Context_for_call_exception
--/
--/ POSTCONDITIONS:
--/ the wait does not occur
--/
--/ no change in priority is registered
--/
--/
--/
--/ and Generic_Process_Attribute_Modifiers;
pragma page;

```

```

./ .....
./ MODULE NAME:
./ process_attribute_modifiers
./
./ MODULE TYPE:
./ package specification
./
./ MODULE PURPOSE:
./ an example of (and a template for) an instantiation of the generic
./ package generic_process_attribute_modifiers
./
./ .....
./ MODULE DESCRIPTION:
./ package instantiating the modification of attributes of already existing
./ processes where:
./ error checking for illegal context for call is enabled
./ error checking for illegal_process_identifier is enabled
./
./ MODULE CONTENTS:
./ process management (package specification)
./ die (procedure)
./ illegal context for call (exception)
./ illegal_process_identifier (exception)
./ kill (procedure)
./ set_process_preemption (procedure)
./ set_process_priority (procedure)
./ wait (overloaded (2) procedure)
./
./ REFERENCES:
./
./ DESIGN DOCUMENTS:
./ none
./
./ USER'S MANUAL:
./ Kernel User's Manual
./
./ TESTING AND VALIDATION:
./ none
./
./ NOTES:
./ none
./
./ .....
./ MODIFICATION HISTORY:
./ 18aug88 bamberg created (complies clean)
./ 29jun89 dvk Bug 0040 SEI: add checks for illegal PIDs
./ 27aug89 bamberg final doc. check for 01sep89 freeze
./
./ .....
./ VERSION: 3 0
./
./ DISCLAIMER
./
./ The following copyright must be included in this software and
./ all software utilizing this software.
./

```

```

./ .....
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....
./
./ with Generic Process Attribute Modifiers;
./ pragma Elaborate (Generic Process Attribute Modifiers);
./ .. to get: see MODULE CONTENTS
./
./ .....
./ package short name: PAM
./ .....
./
./ package Process Attribute Modifiers
./
./ is new
./ Generic Process Attribute Modifiers (
./ Illegal Context For Call Enabled => True,
./ Illegal Process Identifier Enabled => True);
./ pragma (page);

```

```

./ .....
./ MODULE NAME:
./ generic_process_attribute_readers
./
./ MODULE TYPE:
./ generic_package_specification
./
./ MODULE PURPOSE:
./ support the read-only access to certain process attributes
./
./ .....
./ MODULE DESCRIPTION:
./ collection of subprograms to (only) read certain process attributes
./
./ .....
./ MODULE CONTENTS:
./ generic_process_attribute_readers (generic package specification)
./ get_process_preemption (function)
./ get_process_priority (function)
./ illegal_context_for_call (exception)
./ illegal_context_for_call_enabled (generic formal parameter)
./ illegal_process_identifier (exception)
./ illegal_process_identifier_enabled (generic formal parameter)
./ name_of (function)
./ who_am_i (function)
./
./ .....
./ REFERENCES:
./ DESIGN DOCUMENTS:
./ Kernel Facilities Definition
./ -REQ: 5.1.8
./ -PRIM: 10.1.31; 16.1.7; 18.1.2; 18.1.4
./ -TIME: N/A
./
./ .....
./ USER'S MANUAL:
./ Kernel User's Manual
./
./ .....
./ TESTING AND VALIDATION:
./ Unit testing
./
./ .....
./ NOTES:
./ In name of
./ If checking for illegal_process, identifier is disabled, the function
./ will return the empty (zero length) string in the event an illegal
./ process ID is passed in as a parameter.
./ In who_am_i:
./ If checking for illegal_context, call is disabled, the application is
./ asserting that it guarantees that the calling unit is NOT an
./ interrupt handler; if the calling unit IS an interrupt handler, a
./ null process identifier is returned (something must be returned or
./ program error would be raised; a null process identifier is as
./ innocuous as possible, as interrupt handlers are not real processes)
./ In get_process_preemption:
./ If checking for illegal_context, call is disabled, the application is
./ asserting that it guarantees that the calling unit is NOT an
./ interrupt handler; if the calling unit IS an interrupt handler, the
./ value disabled is returned (something must be returned or

```

```

./ program error would be raised; a value of disabled is as innocuous
./ as possible, as an interrupt handler may not be preempted by another
./ Kernel process)
./ In get_process_priority:
./ If checking for illegal_context, call is disabled, the application is
./ asserting that it guarantees that the calling unit is NOT an
./ interrupt handler; if the calling unit IS an interrupt handler, the
./ value priority first is returned (something must be returned or
./ program error would be raised; a value of priority first is as
./ innocuous as possible, as interrupts are processed at higher priority
./ than Kernel processes)
./
./ .....
./ MODIFICATION HISTORY:
./ 18aug88 bamberg created (complies clean)
./ 02nov88 bamberg added NOTES commentary for build 3 implementation
./ 08jun89 .dsk to fix bug report 0040 SEI
./ 27aug89 bamberg final doc. check for 01sep89 freeze
./
./ .....
./ VERSION: 3.0
./
./ .....
./ DISCLAIMER:
./ The following copyright must be included in this software and
./ all software utilizing this software.
./
./ .....
./ DISTRIBUTION:
./ Approved for public release; distribution is unlimited.
./
./ .....
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ .....
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.

```

```

.....
pragma (page);

with Kernel_Exceptions;
-- to get: illegal context for call
-- illegal_process_idenlife;

with Hardware_Interface; use Hardware_Interface;
-- to get: hw_string

with Process_Table;
-- to get: process_idenlife

with Schedule_Types;
-- to get: preemption
-- priority

generic

-- ERROR checking conditional compilation flags
--
--REQ: 5.1.8
Illegal_Context_For_Call_Enabled : In Boolean := True;
Illegal_Process_Idenlife_Enabled : In Boolean := True;
--/.....
--/ package short name: GPAR
--/.....

package Generic_Process_Attribute_Readers
is
package Ptb
renames Process_Table;
package Ke
renames Kernel_Exceptions;
package Si
renames Schedule_Types;
--
-- illegal context for call raised by:
-- who am I
-- get process preemption
-- get process priority
-- PRIM: 16.1.7; 18.1.2; 18.1.4
Illegal_Context_For_Call_exception
renames Ke_Illegal_Context_For_Call;

```

```

.. illegal_process_identifier raised by;
.. name_of
..
.. PRIM: 10.1.31
illegal_process_identifier : exception
renames KeIllegal_Process_Identifier;
pragma (page);

```

```

function Name_Of (
  Process_Id : Pib.Process_Identifier)
  return Pib.String;
--/
--/ SUBPROGRAM NAME:
--/ name_of
--/
--/ DESCRIPTION:
--/ allows a process to obtain the logical (string) name of a process
--/
--/ --REQ: 7.1.24; 7.1.25; 7.1.26; 10.1.1
--/
--/ --PRIM: 16.1.8
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ process_ID
--/ the ID of the process for which the logical name is to be obtained
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the logical (string) name of the process that corresponds to the
--/ process_ID
--/
--/ PRECONDITIONS:
--/ the process_ID must be a legal value
--/
--/ ACTIONS:
--/ return the logical name of the specified process
--/
--/ POSTCONDITIONS:
--/ no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/ when this primitive is called with an undeclared process ID:
--/ the Kernel raises illegal_process_identifier exception
--/ POSTCONDITIONS:
--/ the call is rejected; no meaningful value is returned
--/
--/
pragma (page);

```

```

function Who_Am_I return PtoProcess Identifier;
--/
--/ SUBPROGRAM NAME:
--/ who_am_i
--/
--/ DESCRIPTION:
--/ allows a process to obtain its own process ID
--/
--/ --REQ: 7.1.23
--/ --PRIM: 16.1.7
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ function return value
--/ the Kernel-assigned process ID for this process
--/
--/ PRECONDITIONS:
--/ none (other than the calling process must be alive, but if it
--/ weren't, it couldn't make this call)
--/
--/ ACTIONS:
--/ return the process ID of the calling process
--/
--/ POSTCONDITIONS:
--/ no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in an interrupt handler:
--/ the Kernel raises illegal_context_for_call_exception
--/ POSTCONDITIONS:
--/ the call is rejected; no meaningful value is returned
--/
pragma (page);

```

```

function Get_Process_Preemption return St_Preemption;
--/
--/ SUBPROGRAM NAME:
--/ get_process_preemption
--/
--/ DESCRIPTION:
--/ queries the current value of the preemption status of the calling
--/ process
--/
--/ --REQ: 9.1.13
--/ --PRIM: 18.1.2
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ function return value
--/ the preemption status of the calling process
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ the Kernel returns the value of the preemption status
--/
--/ POSTCONDITIONS:
--/ no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in an interrupt handler:
--/ the Kernel raises illegal_context_for_call_exception
--/ POSTCONDITIONS:
--/ the call is rejected; no meaningful value is returned
--/
pragma (page);

```



```

function Get_Process_Priority return St.Priority;
--/
--/ SUBPROGRAM NAME:
--/   get_process_priority
--/
--/ DESCRIPTION:
--/   queries the current value of the priority of the calling process
--/
--/ --REQ: 9.1.9
--/ --PRIM: 18.1.4
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/   function return value
--/   the current priority of the calling process
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the Kernel returns the value of the priority level
--/
--/ POSTCONDITIONS:
--/   no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/   when this primitive is called in an interrupt handler:
--/   the Kernel raises illegal_context_for_call_exception
--/
--/ POSTCONDITIONS:
--/   the call is rejected; no meaningful value is returned
--/
--/
end Generic_Process_Attribute_Readers;
pragma page;

.....
--/ MODULE NAME:
--/   process_attribute_readers
--/
--/ MODULE TYPE:
--/   package specification
--/
--/ MODULE PURPOSE:
--/   an example of (and a template for) an instantiation of the generic
--/   package generic_process_attribute_readers
--/
--/ MODULE DESCRIPTION:
--/   package instantiating the capabilities to read certain process
--/   attributes where:
--/     error checking for illegal_context_for_call is enabled
--/     error checking for illegal_process_identifier is enabled
--/
--/ MODULE CONTENTS:
--/   process_attribute_readers (package specification)
--/   get_process_preemption (function)
--/   get_process_priority (function)
--/   illegal_context_for_call (exception)
--/   illegal_process_identifier (exception)
--/   name_of (function)
--/   who_am_i (function)
--/
--/ REFERENCES:
--/   DESIGN DOCUMENTS:
--/     none
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   none
--/
--/ NOTES:
--/   none
--/
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (complies clean)
--/   08jun89 dtk added illegal_process_identifier exception
--/           to fix bug report 00-40 SEI
--/   27aug89 bamberg final doc. check for 01sep89 freeze
--/
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/   The following copyright must be included in this software and
--/   all software utilizing this software.

```

```

../ DISTRIBUTION:
../
../ Approved for public release, distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../

```

```

with Generic Process Attribute Readers;
pragma Elaborate (Generic Process Attribute Readers);
.. to get: see MODULE CONTENTS

```

```

../
../ package short name: PAR
../

```

```

package Process Attribute Readers

```

```

is new Generic Process Attribute Readers (
  Illegal Context For Call Enabled => True,
  Illegal Process Identifier Enabled => True);
pragma (page);

```

```

../
../ MODULE NAME:
../ generic_interrupt_globals
../
../ MODULE TYPE:
../ generic package specification
../
../ MODULE PURPOSE:
../ exports interrupt global information
../
../
../ MODULE DESCRIPTION:
../ collection of globals related to interrupt management but which can be
../ used generally by anyone
../
../ MODULE CONTENTS:
../ generic_interrupt_globals (generic package specification)
../ interrupt_condition (type)
../ interrupt_name (type)
../ interrupt_owner (type)
../ interrupt_source (type)
../ interrupt_state (type)
../ interrupt_table (data structure)
../ interrupt_table_entry (type)
../ interrupt_table_type (type)
../ interrupt_vector (data structure)
../ null_handler (constant)
../ null_handler_body (procedure)
../
../ REFERENCES:
../ DESIGN DOCUMENTS:
../ Kernel Facilities Definition
../ --REQ: 11.1.1, 11.1.4, 11.1.5, 11.1.6, 11.1.14, 11.1.17
../ --REQ: 11.1.19, 11.1.21
../ --PRIM: 20.1.1; 20.1.2; 20.1.3
../ --TIME: N/A
../
../ USER'S MANUAL:
../ Kernel User's Manual
../
../ TESTING AND VALIDATION:
../ Integration testing with Mproc and Time_keeper
../
../ NOTES
../ none
../
../
../ MODIFICATION HISTORY:
../ 18aug88 bamberg created (compiles clean)
../ 16oct88 firth revised bind interface
../ 20oct88 firth revised interrupt table
../ 1nov88 firth final cleanup
../ 12jan89 firth final doc. check for beta release (no change)
../ 3apr89 firth made null handler a true handler with no body
../ 19apr89 firth removed obsolete generic formal params

```


--REQ: 11.1.4; 11.1.5
--PRIM: 20.1.1; 20.1.2; 20.1.3

type Interrupt_State is (Enabled, Disabled);

-- this type indicates the source of an Interrupt

-- values include:

Internal - Interrupt was generated by a simulate Interrupt call
External - Interrupt was generated by hardware the the hardware
Interrupt handling mechanism

type Interrupt_Source is (Internal, External);

-- the null Interrupt handler, is a procedure with null body

procedure Null_Handler_Body;

-- this constant designates the null Interrupt handler

Null_Handler : constant Hw_Address := Hw_Address(Null_Handler_Body/Address);

pragma (page);

-- the information maintained about Interrupts includes:

-- owner

-- the owner of the Interrupt

-- condition

-- the condition of the Interrupt

-- Values include:

Bound (a handler has been associated with the named Interrupt
via a Kernel call)

Unbound (a handler has not been associated with the named
Interrupt via a Kernel call)

-- state

-- The state of the Interrupt

-- Values include:

Enabled (the Interrupt is enabled)

Disabled (the Interrupt is disabled)

-- Interrupt source

-- Indication of where the Interrupt sequence originated from

-- Values include:

Internal (the Interrupt was generated by a simulate Interrupt
call)

External (the Interrupt was generated external the processor
as some hardware Interrupt)

-- can preempt

-- after executing the handler, an Interrupt can do one of
two things

a. return control to the Interrupted process

b. return control to the DARK scheduler, which of course
may elect not to resume the Interrupted process

a value False for the attribute 'can_preempt' means that
the Interrupt will always do (a); a value True that it will
always do (b).

-- Interrupt handler

-- Interrupt service routine for the Interrupt

-- tool interface

-- Indication of whether a tool interface has been enabled

-- Values include:

True (tool interface established)

False (tool interface not established)

-- process ID

-- the monitoring process's identifier

type Interrupt_Table Entry is record

```

Owner
Condition
State
Source
Can Preempt
Handler
Tool Interface Enabled
Monitoring_Process_Id
Process_Table_Null_Process;

end record ;
pragma (page);

```

```

-- this is the template for declaring an Interrupt table. It is indexed
-- by the range of Interrupt names
--
-- each Interrupt entry contains all of the information on an Interrupt;

type Interrupt_Table_Type is array (Interrupt_Name) of Interrupt_Table_Entry;
Interrupt_Table : Interrupt_Table_Type;

-- this is the transfer vector actually used by the Interrupt
-- encapsulation routines. Whenever a handler is enabled,
-- its code address is set in this vector. Whenever a handler
-- is disabled (including initially), the vector designates
-- a null handler, ie a procedure with null body

Interrupt_Vector : array (Interrupt_Name) of Hw_Address
:= (others => Null_Handler);

end Generic Interrupt Globals;
pragma (page);

```

```

./ .....
./ MODULE NAME:
./   interrupt_globals
./
./ MODULE TYPE:
./   package specification
./
./ MODULE PURPOSE:
./   an example of (and a template for) an instantiation of the generic
./   package generic_interrupt_globals
./ .....
./ MODULE DESCRIPTION:
./   package instantiating the interrupt_globals (notably the
./   legal_interrupt_table)
./ .....
./ MODULE CONTENTS:
./   interrupt_globals (package specification)
./   interrupt_condition (type)
./   interrupt_name (type)
./   interrupt_source (type)
./   interrupt_state (type)
./   interrupt_table (data structure)
./   interrupt_table_entry (type)
./   interrupt_table_type (type)
./ .....
./ NOTES:
./   none
./ .....
./ MODIFICATION HISTORY:
./   18aug88 bamberg created (compiles clean)
./   11jan89 firh revised doc. for beta release
./   19apr89 firh removed obsolete generic actual params
./   14may89 firh removed old documentation per bug 0071_SEI
./ .....
./ VERSION: 3.0
./ .....
./ DISCLAIMER:
./   The following copyright must be included in this software and
./   all software utilizing this software.
./ .....
./ DISTRIBUTION:
./   Approved for public release; distribution is unlimited.
./ .....
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.

```

```

./ .....
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....
pregame (page);

```

with Generic Interrupt Globals;
pragmas Elaborate (Generic Interrupt Globals);
.. to get: see MODULE CONTENTS

.....
..// package short name: IG

package Interrupt Globals

is new Generic Interrupt Globals;

pragmas (page);

.....
..// MODULE NAME:
..// generic_interrupt_management
..//
..// MODULE TYPE:
..// generic package specification
..//
..// MODULE PURPOSE:
..// export routines needed to bind, enable, disable,
..// and simulate interrupts
..// customize error checking appropriately by generic parameters
..//
..//
..// MODULE DESCRIPTION:
..//
..//
..// GENERIC PARAMETERS
..// illegal_interrupt_enabled (Boolean flag)
..// illegal_interrupt_handler_address_enabled (Boolean flag)
..// no_interrupt_handler_bound_enabled (Boolean flag)
..// replacing_previous_interrupt_handler_enabled (Boolean flag)
..// reserved_interrupt_enabled (Boolean flag)
..//
..// MODULE CONTENTS:
..// Enable (procedure)
..// Disable (procedure)
..// Enabled (function)
..// Simulate Interrupt (procedure)
..// Bind Interrupt_Handler (procedure)
..//
..// REFERENCES:
..// DESIGN DOCUMENTS:
..// Kernel Facilities Definition
..// --REQ: 11.1.15, 11.1.20
..// --PRIM: 20.1.1, 20.1.2, 20.1.4, 20.1.5
..// --TIME: N/A
..//
..// USER'S MANUAL:
..// Kernel User's Manual
..//
..// TESTING AND VALIDATION:
..// Unit test
..// Integration test with standalone handlers
..// Integration test with DARK device handlers
..// Integration test with Application device handlers
..//
..// NOTES:
..// Interrupt management is rather different on Kproc and Nproc.
..// However, the interface is the same, the bodies of the procedures
..// test which processor they are on
..//
..//
..// MODIFICATION HISTORY:
..// 23 Sep 88 created
..// 20 Oct 88 revised Interface

```

--/ 11 Jan 89 final doc. check for beta release
--/ 19 Jan 89 Documentation seriously in error - revised
--/ 25 Jan 89 final check and recompile
--/ 25 Jan 89 final final check and recompile
--/ 22 May 89 revised documentation on 'can preempt' per bug 0051 SEI
--/ 26 May 89 revised documentation on 'disable per bug 0002_SEI'
--/
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
pragme (page);

```

```

with Hardware_Interface use Hardware_Interface;
-- to get: hw_address

with Interrupt_Globals;
-- to get: Interrupt_name

with Kernel_Exceptions;
-- to get: illegal_interrupt
-- to get: illegal_interrupt_handler_address
-- to get: no_interrupt_handler_bound
-- to get: replacing_previous_interrupt_handler
-- to get: reserved_interrupt

generic
-- error checking conditional compilation flags

illegal_interrupt_enabled : In Boolean := True;
illegal_interrupt_handler_address_enabled : In Boolean := True;
no_interrupt_handler_bound_enabled : In Boolean := True;
replacing_previous_interrupt_handler_enabled : In Boolean := True;
reserved_interrupt_enabled : In Boolean := True;

-- .....
--/ package short name: GIM
--/ .....

package Generic_Interrupt_Management
is
package lg
renames Interrupt_Globals;
-- Error Status Codes 20.3
-- illegal_interrupt is intended to be raised by any of
-- enable
-- disable
-- enabled
-- simulate_interrupt
-- bind_interrupt_handler
-- If the interrupt name passed as a parameter does not designate a
-- legal interrupt.
--
-- In the current implementation, all values of the subtype interrupt_name
-- are legal, so this exception cannot be raised.

illegal_interrupt : exception
renames Kernel_Exceptions.illegal_interrupt;

-- illegal_interrupt_handler_address is raised by
-- bind_interrupt_handler
-- if the subprogram address passed as the handler_code parameter

```



```

-- is not a legal code address
--
-- PRIM: 20.1.5
--
Illegal Interrupt_Handler_Address : exception
renames Kernel_Exceptions.Illegal_Interrupt_Handler_Address;

-- no interrupt_handler_bound is raised by
-- enable
-- simulate interrupt
-- if no interrupt handler has been bound
--
-- PRIM: 20.1.1, 20.1.2, 20.1.4

No_Interrupt_Handler_Bound : exception
renames Kernel_Exceptions.No_Interrupt_Handler_Bound;

-- replacing previous_interrupt_handler is raised by
-- bind_interrupt_handler
-- if a handler has previously been bound
--
-- PRIM: 20.1.5

Replacing_Previous_Interrupt_Handler : exception
renames Kernel_Exceptions.Replacing_Previous_Interrupt_Handler;

-- reserved_interrupt is raised by
-- enable
-- disable
-- enabled
-- simulate interrupt
-- bind_interrupt_handler
-- if the interrupt name passed as a parameter designates an interrupt
-- reserved to the Ada runtime or the Kernel.
--
-- PRIM: 20.1.1, 20.1.1.1, 20.1.3, 20.1.4, 20.1.5

Reserved_Interrupt : exception
renames Kernel_Exceptions.Reserved_Interrupt;

pragma (page);

```

```

procedure Enable (
  Interrupt : in lg Interrupt_Name);
--/
--/ SUBPROGRAM NAME:
--/ enable
--/
--/ DESCRIPTION:
--/ allows the specified interrupt to occur
--/
--/ --REQ: 11.1.4
--/ --PRIM: 20.1.1
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Interrupt
--/ the name of the interrupt to be enabled
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ The interrupt is legal
--/ The interrupt is not reserved
--/ A handler has already been bound to the interrupt
--/
--/ ACTIONS:
--/ Indicates that the interrupt is allowed to occur
--/
--/ POSTCONDITIONS:
--/ the interrupt may now occur
--/ the Kernel notes that the interrupt state is enabled
--/
--/ ERR OR PROCESSING:
--/ Legality of the interrupt is checked by the Ada language,
--/ since the type interrupt_globals.interrupt_name embraces
--/ all legal interrupts. The exception Constraint_Error is
--/ raised for an illegal interrupt.
--/
--/ The subprogram checks that the interrupt is free to the
--/ application, and raises Reserved_Interrupt if it is not.
--/
--/ The subprogram checks that a handler has been bound, and
--/ raises No_Interrupt_Handler_Bound if one has not.
--/
--/ If an error occurs but its reporting is disabled, the subprogram
--/ silently returns without changing the state of the interrupt.
--/
--/
pragma (page);

```

```

procedure Disable (
  Interrupt : in lg Interrupt Name);
--=====
--/ SUBPROGRAM NAME:
--/
--/
--/ DESCRIPTION:
--/ ignores the specified Interrupt whenever it occurs
--/
--/ --REQ: 11.1.5
--/ --PRIM: 20.1.2
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Interrupt
--/ the name of the Interrupt to be disabled
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ The Interrupt is legal
--/ The Interrupt is not reserved
--/ NOTE that a handler need not have been bound
--/
--/ ACTIONS
--/ Indicates that the Interrupt is ignored whenever it occurs
--/
--/ POSTCONDITIONS:
--/ the Interrupt is disabled
--/ the Kernel notes that the Interrupt state is disabled
--/
--/ ERROR PROCESSING:
--/ Legality of the Interrupt is checked by the Ada language,
--/ since the type Interrupt_globals.Interrupt_name embraces
--/ all legal Interrupts. The exception Constraint_Error is
--/ raised for an illegal Interrupt.
--/
--/ The subprogram checks that the Interrupt is free to the
--/ application, and raises Reserved Interrupt if it is not.
--/
--/ If an error occurs but its reporting is disabled, the subprogram
--/ silently returns without changing the state of the Interrupt.
--/
pragma (page);

```

```

function Enabled (
  Interrupt : lg Interrupt Name)
return Boolean;
--=====
--/ SUBPROGRAM NAME:
--/ enabled
--/
--/ DESCRIPTION:
--/ queries
--/
--/ --REQ: 11.1.7
--/ --PRIM: 20.1.3
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Interrupt
--/ the name of the Interrupt being queried
--/
--/ default value:
--/ none
--/
--/ function return value
--/ indication of whether or not the specified Interrupt is enabled
--/
--/ values include:
--/ true (the specified Interrupt is enabled)
--/ false (the specified Interrupt is disabled)
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ The Interrupt is legal
--/ The Interrupt is not reserved
--/
--/ ACTIONS:
--/ the Kernel returns an indication of the Interrupt state
--/
--/ POSTCONDITIONS:
--/ no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/ Legality of the Interrupt is checked by the Ada language,
--/ since the type Interrupt_globals.Interrupt_name embraces
--/ all legal Interrupts. The exception Constraint_Error is
--/ raised for an illegal Interrupt.
--/
--/ The subprogram checks that the Interrupt is free to the
--/ application, and raises Reserved Interrupt if it is not.
--/
--/ If an error occurs but its reporting is disabled, the subprogram
--/ returns false, regardless of the actual state of the Interrupt.
--/

```

--/ =====
pragma (page);

procedure Simulate Interrupt (
Interrupt : in lg Interrupt Name);

--/ =====
--/ SUBPROGRAM NAME:

--/ simulate Interrupt

--/ DESCRIPTION:

--/ simulates the occurrence of a specified interrupt in software

--/ --REQ: 11.1.2, 11.1.3, 11.1.8

--/ --PRIM: 20.1.4

--/ --TIME: N/A

--/ PARAMETERS:

--/ Interrupt

--/ the name of the interrupt being simulated

--/ default value:

--/ none

--/ PRECONDITIONS:

--/ The interrupt is legal

--/ The interrupt is not reserved

--/ A handler has already been bound to the interrupt

--/ ACTIONS:

--/ the interrupt handler associated with the specified interrupt name
--/ is executed as though the actual hardware interrupt had occurred

--/ POSTCONDITIONS:

--/ upon completion of the interrupt handler execution,

--/ it can preempt is False;

--/ the interrupted code is resumed

--/ it can preempt is True;

--/ the Scheduler is invoked and selects the process to be resumed

--/ ERROR PROCESSING:

--/ Legality of the interrupt is checked by the Ada language,

--/ since the type interrupt_globals.interrupt_name embraces

--/ all legal interrupts. The exception Constraint_Error is

--/ raised for an illegal interrupt.

--/ The subprogram checks that the interrupt is free to the

--/ application, and raises Reserved Interrupt if it is not.

--/ The subprogram checks that a handler has been bound, and

--/ raises No Interrupt_Handler_Bound if one has not.

--/ If the interrupt handler tries to propagate an exception, the

--/ subprogram simulate_interrupt immediately considers the handler to

--/ have completed, and continues with the normal completion action.

--/ (This is the same as the action taken by a real interrupt)

```

--/ If an error occurs but its reporting is disabled, the subprogram
--/ silently returns without taking any action.
--/

```

```

pragma (page);

```

```

--/ =====
--/ SUBPROGRAM NAME:
--/ bind interrupt_handler
--/
--/ DESCRIPTION:
--/ associates the specified interrupt with the handler code to be
--/ executed when the interrupt occurs
--/
--/ --REQ: 11.1.2; 11.1.9; 11.1.10; 11.1.11; 11.1.13
--/
--/ --PRIM: 20.1.5
--/
--/ --TIME: 11.2.3
--/
--/ PARAMETERS:
--/ Interrupt
--/ the name of the interrupt for which the specified handler code
--/ is to be executed
--/
--/ default value:
--/ none
--/
--/ handler_code
--/ the address of the code to be executed as the interrupt handler;
--/ this code must be an Ada procedure with no parameters or may be
--/ code in any other language that follows the compiler's procedure
--/ entry and exit conventions
--/
--/ default value:
--/ none
--/
--/ can_preempt
--/ indicates whether invocation of the handler can cause
--/ the interrupted process to be rescheduled
--/ upon completion of the interrupt handler execution,
--/ it can preempt is False;
--/ the interrupted code is resumed
--/ if can_preempt is True;
--/ the Scheduler is invoked and selects the process to be resumed
--/
--/ default value
--/ true
--/
--/ PRECONDITIONS:
--/ The interrupt is legal
--/ The interrupt is not reserved
--/ A handler has not already been bound to the interrupt (weak)
--/
--/ ACTIONS:

```

```

--/ constructs the execution environment for this interrupt handler,
--/ including: the local interrupt stack and interrupt entry
--/ and exit code. Note that no dummy call frame or stack plug
--/ is needed, since the handlers will be themselves Kernel code
--/
--/ POSTCONDITIONS:
--/ execution environment created for this interrupt handler
--/ note that this postcondition is established even if the
--/ prior state of the interrupt was Bound, ie any previous
--/ handler will be replaced.
--/
--/ ERROR PROCESSING:
--/ Legality of the interrupt is checked by the Ada language,
--/ since the type interrupt_globals.interrupt_name embraces
--/ all legal interrupts. The exception Constraint_Error is
--/ raised for an illegal interrupt.
--/
--/ The subprogram checks that the interrupt is free to the
--/ application, and raises Reserved_interrupt if it is not.
--/
--/ The subprogram checks that a handler has not already been bound.
--/ If one has, it is REPLACED by the new handler, after which
--/ Replacing_Prevous_interrupt_Handler is raised. The handler
--/ is replaced even if this error report is disabled.
--/
--/ If other error reporting is disabled, the subprogram.. silently
--/ returns without changing the state of the interrupt.
--/
--/ NOTES:
--/ The exception Insufficient_Space cannot be raised by this
--/ subprogram. All data structures are static, and all real
--/ interrupts execute on a single stack that is allocated at
--/ program load time. Binding a handler requires no further
--/ allocation of space.
--/
--/ .....
end Generic_interrupt_Management;
pragma page;

--/ .....
--/ MODULE NAME:
--/ interrupt_management
--/
--/ MODULE TYPE:
--/ package specification
--/
--/ MODULE PURPOSE:
--/ an example of (and a template for) an instantiation of the generic
--/ package generic_interrupt_management
--/
--/ .....
--/ MODULE DESCRIPTION:
--/ package instantiating the capabilities to create the interrupt
--/ execution environment and to manipulate interrupts where:
--/ error checking for illegal_interrupt is enabled
--/ error checking for illegal_interrupt_handler_address is enabled
--/ error checking for insufficient_space is enabled
--/ error checking for no_interrupt_handler_bound is enabled
--/ error checking for replacing_previous_interrupt_handler is enabled
--/ error checking for reserved_interrupt is enabled
--/
--/ MODULE CONTENTS:
--/ interrupt_management (generic package specification)
--/ bind_interrupt_handler (procedure)
--/ disable (procedure)
--/ enable (procedure)
--/ enabled (function)
--/ illegal_interrupt (exception)
--/ illegal_interrupt_handler_address (exception)
--/ insufficient_space (exception)
--/ no_interrupt_handler_bound (exception)
--/ replacing_previous_interrupt_handler (exception)
--/ reserved_interrupt (exception)
--/ simulate_interrupt (procedure)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ Kernel Facilities Definition
--/
--/ USER'S MANUAL:
--/ Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/ none
--/
--/ NOTES:
--/ none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (complies clean)
--/ 12jan89 ltrh final revision for beta release
--/

```

```

.../ VERSION 3.0
.../
.../ DISCLAIMER:
.../
.../ The following copyright must be included in this software and
.../ all software utilizing this software.
.../
.../ DISTRIBUTION:
.../
.../ Approved for public release, distribution is unlimited.
.../
.../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
.../ PA. The Software Engineering Institute (SEI) is a federally
.../ funded research and development center established and operated
.../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
.../ Department of Defense under contract F19628-85-C-0003, the SEI
.../ is supported by the services and defense agencies, with the U.S.
.../ Air Force as the executive contracting agent.
.../
.../ Permission to use, copy, modify, or distribute this software and
.../ its documentation for any purpose and without fee is hereby
.../ granted, provided that the above copyright notice appear in all
.../ copies and that both that copyright notice and this permission
.../ notice appear in supporting documentation. Further, the names
.../ Software Engineering Institute or Carnegie Mellon University may
.../ not be used in advertising or publicity pertaining to
.../ distribution of the software without specific, written prior
.../ permission. CMU makes no claims or representations about the
.../ suitability of this software for any purpose. This software is
.../ provided "as is" and no warranty, express or implied, is made
.../ by the SEI or CMU, as to the accuracy and functioning of the
.../ program and related program material, nor shall the fact of
.../ distribution constitute any such warranty. No responsibility is
.../ assumed by the SEI or CMU in connection herewith.
.../
.../
.../ with Generic Interrupt Management;
.../ to get: see MODULE CONTENTS
.../
.../
.../ package short name: IM
.../
.../ package Interrupt_Management

Is new Generic Interrupt Management (
Illegal Interrupt Enabled => True,
Illegal Interrupt Handler Address Enabled => True,
No Interrupt Handler Bound Enabled => True,
Replacing Previous Interrupt Handler Enabled => True,
Reserved Interrupt_Enabled => True);

prgms (page);

```

```

.../
.../ MODULE NAME:
.../ generic_semaphore_management
.../
.../ MODULE TYPE:
.../ generic_package_specification
.../
.../ MODULE PURPOSE:
.../ provide the abstraction of classical (Dijkstra) semaphores to control
.../ process synchronization and mutual exclusion
.../
.../
.../ MODULE DESCRIPTION:
.../ collection of subprograms to manipulate semaphores (i.e., process
.../ queues)
.../
.../ MODULE CONTENTS:
.../ generic_semaphore_management (generic_package_specification)
.../ claim (overloaded (3) procedure)
.../ claim_lined_out (exception)
.../ claim_lined_out_enabled (generic_formal_parameter)
.../ illegal_context_for_call (exception)
.../ illegal_context_for_call_enabled (generic_formal_parameter)
.../ not_my_semaphore (exception)
.../ not_my_semaphore_enabled (generic_formal_parameter)
.../ release (procedure)
.../
.../ REFERENCES:
.../ DESIGN DOCUMENTS:
.../ Kernel Facilities Definition
.../ REQ: 5.1.8; 8.1.1; 8.1.2; 8.1.14
.../ PRIM: 17.1.2; 17.1.3
.../ TIME: 8.2.1
.../
.../ USER'S MANUAL:
.../ Kernel User's Manual
.../
.../ TESTING AND VALIDATION:
.../ Unit testing
.../
.../ NOTES:
.../ none
.../
.../
.../ MODIFICATION HISTORY:
.../ 18aug88 bamberg created (complies clean)
.../ 13jan88 firh final doc. changes for beta release
.../
.../
.../ VERSION: 3.0
.../
.../ DISCLAIMER:
.../ The following copyright must be included in this software and
.../ all software utilizing this software

```

```

--/
--/ DISTRIBUTION:
--/
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
pragma (page);

```

```

with Kernel_Exceptions:
  .. to get: claim_timed_out
  .. illegal_context_for_call
  .. not_my_semaphore

with Process_Table:
  .. to get: semaphore

with Schedule_Types:
  .. to get: current_process_priority
  .. priority

with Time_Globals:
  .. to get: elapsed_time
  .. epoch_time

generic
  .. ERROR checking conditional compilation flags
  ..
  .. REQ: 5.1.8
  ..
  Claim_Timed_Out_Enabled := True;
  Illegal_Context_For_Call_Enabled := True;
  Not_My_Semaphore_Enabled := True;
  Pragma(Page);
  .. /package short name: GMM
  .. .....

package Generic_Semaphore_Management
is
  ..
  .. claim_timed_out_raised_by:
  .. claim
  .. PRIM: 17.1.2
  ..
  Claim_Timed_Out : exception
  renames Kernel_Exceptions.Claim_Timed_Out;
  ..
  .. illegal_context_for_call_raised_by:
  .. claim
  .. release
  .. PRIM: 17.1.2, 17.1.3
  ..
  Illegal_Context_For_Call : exception

```

```

renames Kernel_Exceptions.Illegal_Context_For_Call;
-- not my semaphore raised by:
-- release
-- PRIM: 17.1.3
--
Not My Semaphore : exception
renames Kernel_Exceptions.Not_My_Semaphore;
pragma (page);

```

```

procedure Claim (
  Semaphore_Name : In Process_Table.Semaphore;
  Resumption_Priority : In Schedule.Types.Priority :=
    Schedule.Types.Current_Process_Priority);
--
-- SUBPROGRAM NAME:
-- claim
--
-- DESCRIPTION:
-- allows a process to try to claim a semaphore, blocking until the
-- semaphore is claimed
--
-- REQ: 8.1.3; 8.1.4; 8.1.7; 8.1.8; 8.1.9; 8.1.16;
-- REQ: 9.1.6; 9.1.8
-- PRIM: 17.1.2
-- TIME: 8.2.2
--
-- PARAMETERS:
-- semaphore_name
-- a semaphore object created by a declaration using type
-- semaphore
-- default value:
-- none
--
-- resumption_priority
-- the priority to be assigned to this process when it becomes
-- unblocked
--
-- default value:
-- the calling process's current priority
--
-- PRECONDITIONS:
-- the semaphore has been declared
--
-- ACTIONS:
-- the process registers a request to claim a semaphore
-- if the semaphore is free, the process claims the semaphore;
-- otherwise, the calling process blocks (if permitted, see ERROR
-- PROCESSING below) until the semaphore becomes free
-- the kernel possibly changes the process priority
--
-- POSTCONDITIONS:
-- the semaphore is not free
-- the priority of this process is its resumption priority
-- if the semaphore was free when the process claimed it, then the
-- process has control of the semaphore, the process is either
-- suspended or running depending on the discretion of the kernel
-- scheduler, otherwise (if the process was blocked waiting for the
-- semaphore to become free) the process becomes unblocked and is
-- either suspended or running depending on the discretion of the
-- kernel scheduler
--

```



```

--/ ERROR PROCESSING:
--/ when claim is called within an interrupt handler:
--/ the kernel raises illegal context for call
--/ POSTCONDITIONS:
--/ the call is rejected; no attempt to claim a semaphore is noted;
--/ no change in priority is registered
--/
pragma (page);

```

```

procedure Claim (
  Semaphore Name : in Process Table Semaphore;
  Within Elapsed Time : in Time Globals Elapsed Time;
  Resumption Priority : in Schedule Types Priority :=
    Schedule.Types.Current.Process.Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ claim
--/
--/ DESCRIPTION:
--/ allows a process to try to claim a semaphore within a specified
--/ elapsed time
--/ --REQ: 8.1.3; 8.1.4; 8.1.5; 8.1.7; 8.1.8; 8.1.9; 8.1.16
--/ --REQ: 9.1.6; 9.1.8
--/ --PRIM: 17.1.2
--/ --TIME: 8.2.2
--/ PARAMETERS:
--/ semaphore name
--/ a semaphore object created by a declaration using type
--/ semaphore
--/ default value:
--/ none
--/ within elapsed time
--/ duration within which the request to claim a semaphore must be
--/ honored or the request is withdrawn
--/ default value:
--/ none
--/ resumption priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked
--/ default value:
--/ the calling process's current priority
--/ PRECONDITIONS:
--/ the semaphore has been declared
--/
--/ ACTIONS:
--/ the process registers a request to claim a semaphore by a specified
--/ elapsed time
--/ if the semaphore is free, the process claims the semaphore;
--/ otherwise, the calling process blocks (if permitted, see ERROR
--/ PROCESSING below) until the semaphore becomes free or until the
--/ specified timeout expires (for the latter case, see ERROR
--/ PROCESSING below)
--/ if the semaphore is not free and the timeout is for zero or less
--/ elapsed time, then the process does not wait for the semaphore to

```

```

.. become free (see ERROR PROCESSING below)
.. the Kernel possibly changes the process priority
..
.. / POSTCONDITIONS:
.. the semaphore is not free
.. the priority of this process is its resumption priority
.. If the semaphore was free when the process claimed, then the
.. process has control of the semaphore, the process is either
.. suspended or running depending on the discretion of the Kernel
.. Scheduler; otherwise (if the process was blocked waiting for the
.. semaphore to become free) the process becomes unblocked and is
.. either suspended or running depending on the discretion of the
.. Kernel Scheduler
..
.. / ERROR PROCESSING:
.. when the claim timeout expires:
.. the Kt:mel raises claim_timeout exception
.. POSTCONDITIONS:
.. no claim is done (the claim request cannot be satisfied within
.. the specified elapsed time)
.. the priority of this process is its resumption priority
.. when claim is called within an interrupt handler:
.. the Kernel raises illegal_context_for_call
.. POSTCONDITIONS:
.. the call is rejected; no attempt to claim a semaphore is noted;
.. no change in priority is registered
..
.. =====
pragma (page);

```

```

procedure Claim (
  Semaphore Name : in Process Table Semaphore;
  By Epoch Time : in Time Globals Epoch Time;
  Resumption Priority : in Schedule Types Priority :=
    Schedule Types Current Process Priority);
  -----
  -- / SUBPROGRAM NAME:
  -- claim
  --
  -- / DESCRIPTION:
  -- allows a process to try to claim a semaphore by a specified epoch
  -- time
  --
  -- --REQ: 8.1.3, 8.1.4, 8.1.6, 8.1.7, 8.1.8, 8.1.9, 8.1.16;
  -- --REQ: 9.1.6, 9.1.8
  --
  -- --PRIM: 17.1.2
  --
  -- --TIME: 8.2.2
  --
  -- / PARAMETERS:
  -- semaphore name
  -- a semaphore object created by a declaration using type
  -- semaphore
  --
  -- default value:
  -- none
  --
  -- by epoch, time
  -- absolute time by which the request to claim a semaphore must be
  -- honored or the request is withdrawn
  --
  -- default value:
  -- none
  --
  -- resumption priority
  -- the priority to be assigned to this process when it becomes
  -- unblocked
  --
  -- default value:
  -- the calling process's current priority
  --
  -- / PRECONDITIONS:
  -- the semaphore has been declared
  --
  -- / ACTIONS:
  -- the process registers a request to claim a semaphore by a specified
  -- epoch (absolute) time
  -- If the semaphore is free, the process claims the semaphore;
  -- otherwise, the calling process blocks (if permitted, see ERROR
  -- PROCESSING below) until the semaphore becomes free or until the
  -- specified timeout expires (for the latter case, see ERROR
  -- PROCESSING below)
  -- If the semaphore is not free and the timeout is for a non future
  -- epoch time, then the process does not wait for the semaphore to

```

```

.. become free (see ERROR PROCESSING below)
.. the Kernel possibly changes the process priority
..
.. POSTCONDITIONS:
.. the semaphore is not free
.. the priority of this process is its resumption priority
.. if the semaphore was free when the process claimed, then the
.. process has control of the semaphore, the process is either
.. suspended or running depending on the discretion of the Kernel
.. Scheduler, otherwise if the process was blocked waiting for the
.. semaphore to become free the process becomes unblocked and is
.. either suspended or running depending on the discretion of the
.. Kernel Scheduler
..
.. ERROR PROCESSING:
.. when the claim timeout expires:
.. the Kernel raises claim_timeout_exception
.. POSTCONDITIONS:
.. no claim is done (the claim request cannot be satisfied before
.. the specified epoch time)
.. the priority of this process is its resumption priority
.. when claim is called within an interrupt handler:
.. the Kernel raises illegal_context_for_call
.. POSTCONDITIONS:
.. the call is rejected; no attempt to claim a semaphore is noted;
.. no change in priority is registered
..
.. =====
pragma (page);

```

```

procedure Release (
  Semaphore Name : In Process Table.Semaphore);
..
.. SUBPROGRAM NAME:
.. release
..
.. DESCRIPTION:
.. allows a process to release a semaphore if currently controls
..
.. REQ: 8.1.10; 8.1.11; 8.1.13; 8.1.15
..
.. PRM: 17.1.3
..
.. TIME: 8.2.3
..
.. PARAMETERS:
.. semaphore_name
.. a semaphore object created by a declaration using type
.. semaphore
..
.. default value:
.. none
..
.. PRECONDITIONS:
.. the semaphore has been declared
.. the semaphore has been claimed by the calling process
..
.. ACTIONS:
.. the Kernel releases the semaphore (i.e., makes it available to the
.. next process in the waiting queue)
..
.. POSTCONDITIONS:
.. the semaphore is no longer held by the calling process
.. if there are no other processes in the waiting queue for this
.. semaphore, the semaphore becomes free
.. if there are processes in the waiting queue, then the process at the
.. head of the queue is given the semaphore
..
.. ERROR PROCESSING:
.. when release is called within an interrupt handler:
.. the Kernel raises illegal_context_for_call
.. POSTCONDITIONS:
.. the call is rejected; no attempt to release a semaphore is noted;
.. no change in priority is registered
.. when a request to release a semaphore is made by a process that does
.. not own the semaphore:
.. the Kernel raises not_my_semaphore_exception
.. POSTCONDITIONS:
.. the requested action does not take place
..
.. =====

```

end Generic Semaphore Management;
pragma (page);

```

.....
MODULE NAME
 semaphore management

MODULE TYPE
 package specification

MODULE PURPOSE
 an example of (and a template for) an instantiation of the generic
 package generic_semaphore_management

MODULE DESCRIPTION
 package instantiating the capabilities to manipulate semaphores
 (i.e., process queues) where
 error checking for claim timed out is enabled
 error checking for illegal context for call is enabled
 error checking for not my_semaphore is enabled

MODULE CONTENTS
 semaphore management (package specification)
 claim (overloaded (3) procedure)
 claim timed out (exception)
 illegal context for call (exception)
 not my_semaphore (exception)
 release (procedure)

REFERENCES
 DESIGN DOCUMENTS:
 none

USER'S MANUAL
 Kernel User's Manual

TESTING AND VALIDATION:
 none

NOTES
 none

MODIFICATION HISTORY:
 18aug88 bamberg created (complies clean)
 13jan89 fifth final doc. check for beta release

VERSION: 3.0

DISCLAIMER:
 The following copyright must be included in this software and
 all software utilizing this software.

DISTRIBUTION:

```

```

Approved for public release; distribution is unlimited

Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
PA The Software Engineering Institute (SEI) is a federally
funded research and development center established and operated
by Carnegie Mellon University (CMU). Sponsored by the U.S.
Department of Defense under contract F19628-85-C-0003, the SEI
is supported by the services and defense agencies, with the U.S.
Air Force as the executive contracting agent.

Permission to use, copy, modify, or distribute this software and
its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission
notice appear in supporting documentation. Further, the names
of Software Engineering Institute or Carnegie Mellon University may
not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission. CMU makes no claims or representations about the
suitability of this software for any purpose. This software is
provided "as is" and no warranty, express or implied, is made
by the SEI or CMU, as to the accuracy and functioning of the
program and related program material, nor shall the fact of
distribution constitute any such warranty. No responsibility is
assumed by the SEI or CMU in connection herewith.
.....
pragma (page);

```

```

le new Generic_Semaphore_Management (
    Claim Timed Out Enabled => True,
    Illegal Context For Call Enabled => True,
    Not My Semaphore Enabled => True);
pragma page;

```

December 1989

```

--/ all software utilizing this software
--/
--/ DISTRIBUTION:
--/
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
pragme (page):

with Kernel_Exceptions;
-- to get: alarm_expired
-- illegal context for call
-- no alarm set
-- resetting existing alarm

with Schedule_Types;
-- to get: current_process_priority
-- priority

with Time_Globals;
-- to get: elapsed_time
-- epoch_time

generic
--
-- ERROR checking conditional compilation flags
-- REQ: 5.1.8
--
Illegal_Context_For_Call_Enabled : in Boolean := True;
No_Alarm_Set_Enabled : in Boolean := True;
Resetting_Existing_Alarm_Enabled : in Boolean := True;
-- .....
-- package short name: GAM
-- .....
package Generic_Alarm_Management
is
package Ke
renames Kernel_Exceptions;
package St
renames Schedule_Types;
package Tg
renames Time_Globals;
-- REQ: 13.1.4; 13.1.5; 13.2.3
-- REQ: 13.1.1; 13.1.4; 13.1.5
-- PRIM: 22; 22.1.1
Alarm_Expired : exception
renames Ke Alarm_Expired;

```

```

-- illegal context for call raised by:
--   cancel_alarm
--   set_alarm
-- PRIM: 22.1.1; 22.1.2
--
Illegal Context For Call : exception
renames Ke_Illegal_Context_For_Call;
--
-- no_alarm set raised by:
--   cancel_alarm
-- PRIM: 22.1.1; 22.1.2
--
No Alarm Set : exception
renames Ke_No_Alarm_Set;
--
-- resetting_existing_alarm raised by:
--   set_alarm
-- PRIM: 22.1.1
--
Resetting Existing Alarm : exception
renames Ke_Resetting_Existing_Alarm;
pragma (page);

```

```

procedure Set_Alarm (
  Alter_Elapsed_Time : In Tg_Elapsed_Time;
  Expiration_Priority : In St_Priority := St_Current_Process_Priority);
-- /
-- / SUBPROGRAM NAME:
-- / set_alarm
-- /
-- / DESCRIPTION:
-- / defines an alarm that interrupts the calling process if it expires
-- /
-- / REQ: 9.1.6; 9.1.8; 9.1.22
-- / REQ: 13.1.1; 13.1.2; 13.1.5; 13.1.6; 13.1.7; 13.1.8
-- / PRIM: 22.1.1
-- / TIME: 13.2.1
-- /
-- / PARAMETERS:
-- / alter_elapsed_time
-- / the elapsed time immediately after which the alarm_expired
-- / exception is raised
-- / default value:
-- / none
-- /
-- / PRECONDITIONS:
-- / NONE
-- /
-- / ACTIONS:
-- / an alarm event is registered with the Kernel
-- / the Kernel sets its internal timer to note when the specified
-- / elapsed time expires
-- / if the alarm expires, the Kernel raises alarm_expired exception in
-- / the calling process, thereby causing an enforced transfer of
-- / control to a user-provided exception handler and a possible
-- / change in process priority
-- / if the alarm becomes cancelled, there is no transfer or change of
-- / priority
-- /
-- / POSTCONDITIONS:
-- / the alarm event is registered with the Kernel
-- / if the alarm expires:
-- / the priority of this process is its expiration priority
-- / control is transferred to a user provided handler, and the process
-- / state is either suspended or running depending on the
-- / discretion of the Kernel Scheduler
-- / if the alarm becomes cancelled, the internal timer is deactivated
-- /
-- / ERROR PROCESSING:
-- / when set_alarm is called within an interrupt handler:
-- / the Kernel raises illegal_context_for_call
-- / POSTCONDITIONS:
-- / the call is rejected; no alarm is set
-- / when an active alarm is reset:
-- / the Kernel raises resetting_existing_alarm exception

```

```

--/ POSTCONDITIONS:
--/   the requested action takes place
--/

```

pragma (page);

```

procedure Set Alarm (
  For Epoch Time : in Tg.Epoch Time;
  Expiration Priority : in St.Priority := St.Current.Process.Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ set alarm
--/
--/ DESCRIPTION:
--/   defines an alarm that interrupts the calling process if it expires
--/
--/ --REQ: 9.1.6; 9.1.8; 9.1.22
--/ --REQ: 13.1.1; 13.1.3; 13.1.5; 13.1.6; 13.1.7; 13.1.9
--/
--/ --PRIM: 22.1.1
--/
--/ --TIME: 13.2.1
--/
--/ PARAMETERS:
--/   for epoch time
--/   the epoch time immediately at which the alarm_expired
--/   exception is raised
--/
--/   default value:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   an alarm event is registered with the Kernel
--/   the Kernel sets its internal timer to note when the specified
--/   epoch time arrives
--/
--/   If the alarm expires, the Kernel raises alarm_expired exception in
--/   the calling process, thereby causing an enforced transfer of
--/   control to a user provided exception handler and a possible
--/   change in process priority
--/
--/   If the alarm becomes cancelled, there is no transfer or change of
--/   priority
--/
--/ POSTCONDITIONS:
--/   the alarm event is registered with the Kernel
--/
--/   If the alarm expires:
--/   the priority of this process is its expiration priority
--/   control is transferred to a user provided handler, and the process
--/   state is either suspended or running depending on the
--/   discretion of the Kernel Scheduler
--/
--/   If the alarm becomes cancelled, the internal timer is deactivated
--/
--/ ERROR PROCESSING:
--/   when set_alarm is called within an interrupt handler,
--/   the Kernel raises illegal_context for call
--/
--/ POSTCONDITIONS:
--/   the call is rejected, no alarm is set
--/   when an active alarm is reset:
--/   the Kernel raises resetting_existing_alarm_exception
--/

```



```

--/ POSTCONDITIONS:
--/   the requested action takes place
--/

```

pragma (page);

```

procedure Cancel_Alarm;
--/ =====
--/ SUBPROGRAM NAME:
--/   cancel_alarm
--/
--/ DESCRIPTION:
--/   turns off an active alarm (i.e., one that was set but has not yet
--/   expired)
--/
--/   --REQ: 9.1.22; 13.1.10
--/
--/   --PRIM: 22.1.2
--/
--/   --TIME: 13.2.2
--/
--/ PARAMETERS:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the Kernel deactivates the timer
--/
--/ POSTCONDITIONS:
--/   no alarm event exists (any longer) for the calling process
--/
--/ ERROR PROCESSING:
--/   when cancel_alarm is called within an interrupt handler:
--/   the Kernel raises illegal_context_for_call
--/   POSTCONDITIONS:
--/   the call is rejected; no alarm is cancelled
--/   when there is no alarm set for the calling process:
--/   the Kernel raises no_alarm_set_exception
--/   POSTCONDITIONS:
--/   the requested action does not take place
--/
--/ =====
end Generic_Alarm_Management;
pragma (page);

```

```

.....
../ MODULE NAME:
../ alarm_management
../
../ MODULE TYPE:
../ package_specification
../
../ MODULE PURPOSE:
../ an example of (and a template for) an instantiation of the generic
../ package generic_alarm_management
../
../
../ MODULE DESCRIPTION:
../ package instantiating the capabilities to set and cancel an alarm and to
../ report the expiration of an alarm where:
../ error checking for no_alarm_set is enabled
../ error checking for resetting_existing_alarm is enabled
../
../ MODULE CONTENTS:
../ alarm_management(package_specification)
../ alarm_expired(exception)
../ cancel_alarm(procedure)
../ illegal_context_for_call(exception)
../ no_alarm_set(exception)
../ resetting_existing_alarm(exception)
../ set_alarm(overloaded (2) procedure)
../
../ REFERENCES:
../ DESIGN DOCUMENTS:
../ none
../
../ USERS MANUAL:
../ Kernel User's Manual
../
../ TESTING AND VALIDATION:
../ none
../
../ NOTES:
../ the exception: alarm_expired is raised when an alarm expires; when
../ using alarms, the user should take care that the alarm_expired is not
../ masked (or mishandled) by a "when others" clause in an exception
../ handler
../
../ MODIFICATION HISTORY:
../ 18aug88 banberg created (compiles clean)
../ 02aug89 okk Added pragma elaborate to make VMS happy
../
../ VERSION: 3.0
../
../ DISCLAIMER:
../
../ The following copyright must be included in this software and
../ all software utilizing this software.

```

```

../
../ DISTRIBUTION:
../
../ Approved for public release; distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ of Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../ .....
with Generic Alarm Management;
.. to get: see MODULE CONTENTS
pragma Elaborate (Generic Alarm Management);
../ .....
../ package short name: AM
../ .....
package Alarm_Management
is new Generic Alarm_Management (
  illegal_Context For Call Enabled => True,
  No Alarm Set Enabled => True,
  Resetting Existing Alarm Enabled => True);
pragma (page);

```

```

./ .....
./ MODULE NAME:
./   generic time management
./
./ MODULE TYPE:
./   generic package specification
./
./ MODULE PURPOSE:
./   provide the abstraction of Kernel time (i.e., SLICE) and the capability
./   of setting/resetting epoch and elapsed time
./
./ .....
./ MODULE DESCRIPTION:
./   collection of constants and subprograms to define, modify, and read
./   Kernel time
./
./ MODULE CONTENTS:
./   generic time management (generic package specification)
./   adjust elapsed time (procedure)
./   change results in negative elapsed time (exception)
./   change results in negative elapsed time_enabled
./   (generic formal parameter)
./   change results in negative epoch time (exception)
./   change results in negative epoch time_enabled
./   (generic formal parameter)
./   network failure (exception)
./   network failure enabled (generic formal parameter)
./   OK but time already passed (exception)
./   OK but time already passed_enabled (generic formal parameter)
./   read clock (function)
./   synchronize (overloaded (2) procedure)
./   synchronization in progress (exception)
./   synchronization in progress_enabled (generic formal parameter)
./   synchronization_timeout (exception)
./   synchronization_timeout_enabled (generic formal parameter)
./
./ REFERENCES:
./   DESIGN DOCUMENTS:
./   --REQ 5.1.8: 6.1.6
./   --REQ 12.1.2, 12.1.3, 12.1.7, 12.1.8, 12.1.9, 12.1.11
./   --REQ 12.1.13, 12.1.14
./   --PRIM 21.1.3, 21.1.4, 21.1.6
./   --TIME N/A
./
./ USER'S MANUAL:
./   Kernel User's Manual
./
./ TESTING AND VALIDATION:
./   Unit testing
./
./ NOTES:
./   Network failure will never be raised in the current implementation.
./   It is left in the code as place holder for future expansion.

```

```

./ .....
./ MODIFICATION HISTORY:
./   18aug88 bamberg created (complies clean)
./   16feb89 firh corrected error in documentation
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./   The following copyright must be included in this software and
./   all software utilizing this software.
./
./ DISTRIBUTION:
./   Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....
./ pragma (page);

```

```

with Kernel Exceptions;
  .. to get: change_results_in_negative_elapsed_time
  .. change_results_in_negative_epoch_time
  .. network_failure
  .. OK_but_time_already_passed
  .. synchronization_in_progress
  .. synchronization_timeout_expired

with Schedule_Types;
  .. to get: current_process_priority
  .. priority

with Time_Globals;
  .. to get: elapsed_time
  .. epoch_time

generic
  .. error checking conditional compilation flags
  .. REQ: 5.1.8

Change Results In Negative Elapsed Time Enabled : In Boolean := True;
Change Results In Negative Epoch Time Enabled : In Boolean := True;
Network Failure Enabled : In Boolean := True;
Ok But Time Already Passed Enabled : In Boolean := True;
Synchronization In Progress Enabled : In Boolean := True;
Synchronization Timeout Enabled : In Boolean := True;

Pragma(Page);
../ .....
../ package short name: GTM
../ .....

package Generic_Time_Management
is
  .. REQ: 6.1.6; 12.1.2; 12.1.3; 12.1.7; 12.1.8; 12.1.9; 12.1.11
  .. REQ: 12.1.13; 12.1.14

  .. change_results_in_negative_elapsed_time raised by:
  .. adjust_elapsed_time
  .. PRIM: 21.1.3

Change Results In Negative Elapsed Time : exception
renames Kernel_Exceptions.Change_Results_In_Negative_Elapsed_Time;

  .. change_results_in_negative_epoch_time raised by:

```

```

  .. adjust_epoch_time
  .. PRIM: 21.1.4

Change Results In Negative Epoch Time : exception
renames Kernel_Exceptions.Change_Results_In_Negative_Epoch_Time;

  .. network_failure raised by:
  .. synchronize
  .. PRIM: 21.1.6

Network Failure : exception
renames Kernel_Exceptions.Network_Failure;

  .. OK_but_time_already_passed raised by:
  .. adjust_epoch_time
  .. PRIM: 21.1.4

Ok But Time Already Passed : exception
renames Kernel_Exceptions.Ok_But_Time_Already_Passed;

  .. synchronization_in_progress raised by:
  .. synchronize
  .. PRIM: 21.1.6

Synchronization In Progress : exception
renames Kernel_Exceptions.Synchronization_In_Progress;

  .. synchronization_timeout raised by:
  .. synchronize
  .. PRIM: 21.1.6

Synchronization Timeout : exception
renames Kernel_Exceptions.Synchronization_Timeout_Expired;

pragma (page);

```

```

procedure Adjust_Elapsed_Time (
  Adjusted_Time : In Time_Globals.Elapsed_Time);
--/
--/ SUBPROGRAM NAME:
--/   adjust_elapsed_time
--/
--/ DESCRIPTION:
--/   allows the application to increment or decrement the current local
--/   elapsed time by a specified amount of elapsed time
--/
--/ --REQ: 12.1.10
--/
--/ --PRIM: 21.1.3
--/
--/ --TIME: 12.2.1
--/
--/ PARAMETERS:
--/   adjustment
--/   the amount of elapsed time by which the elapsed time should be
--/   advanced (if adjustment is greater than or equal to zero) or
--/   backed up (if adjustment is less than zero)
--/
--/   default value:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the elapsed time counter is adjusted (advanced or backed up) by the
--/   specified amount of elapsed time
--/
--/ POSTCONDITIONS:
--/   the elapsed time counter is adjusted
--/   this affects ALL events pending a time-triggered action - both
--/   elapsed time and epoch time
--/
--/ ERROR PROCESSING:
--/   when the requested change results in negative elapsed time since
--/   start up:
--/   the kernel raises change_results in negative_elapsed_time
--/   exception
--/ POSTCONDITIONS:
--/   the requested change does not occur
--/
--/
--/

```

pragma (page);

```

procedure Adjust_Epoch_Time (
  New_Epoch_Time : In Time_Globals.Epoch_Time);
--/
--/ SUBPROGRAM NAME:
--/   adjust_epoch_time
--/
--/ DESCRIPTION:
--/   allows the application to set the current local epoch time to a
--/   specified new epoch time
--/
--/ --REQ: 12.1.12
--/
--/ --PRIM: 21.1.4
--/
--/ --TIME: 12.2.2
--/
--/ PARAMETERS:
--/   new_epoch_time
--/   the new value for the local epoch time
--/
--/   default value:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the epoch time counter is set to the specified new_epoch_time
--/
--/ POSTCONDITIONS:
--/   the epoch time counter is adjusted
--/   this affects ONLY those events pending an epoch-time-triggered action
--/
--/ ERROR PROCESSING:
--/   when the requested change results in a negative epoch time:
--/   the kernel raises change_results in negative_epoch_time exception
--/ POSTCONDITIONS:
--/   the requested change does not occur
--/   when the requested change results in a backward flow of time:
--/   the kernel raises OK_but_time_already_passed exception
--/ POSTCONDITIONS:
--/   the change occurs as requested
--/
--/
--/

```

pragma (page);

```

function Read_Clock    return Time_Globals.Epoch_Time;
--/
--/ SUBPROGRAM NAME:
--/ read_clock
--/
--/ DESCRIPTION:
--/ provides the current epoch time as computed by the local
--/ processor clock
--/
--/ --REQ: 12.1.15
--/
--/ --PRIM: 21.1.5
--/
--/ --TIME: 12.2.3
--/
--/ PARAMETERS:
--/ function return value
--/ the current epoch time
--/
--/ PRECONDITIONS:
--/ processor initialization has already occurred
--/
--/ ACTIONS:
--/ returns the current epoch time
--/
--/ POSTCONDITIONS:
--/ no change in Kernel data structures
--/
--/ ERROR PROCESSING:
--/ none
--/

```

pragma (page);

```

procedure Synchronize (
  Resumption_Priority : in Schedule_Types.Priority :=
    Schedule_Types.Current_Process_Priority);
--/
--/ SUBPROGRAM NAME:
--/ synchronize
--/
--/ DESCRIPTION:
--/ forces all local processor clocks on Kernel devices to synchronize
--/ time with the local clock on the invoking processor
--/
--/ --REQ: 12.1.16, 12.1.17, 12.1.20, 12.1.21;
--/
--/ --PRIM: 21.1.6
--/
--/ --TIME: 12.2.4, 12.2.5
--/
--/ PARAMETERS:
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked
--/
--/ default value:
--/ the calling process's current priority
--/
--/ PRECONDITIONS:
--/ there is no call to synchronize currently executing
--/
--/ ACTIONS:
--/ the calling processor indicates that all network traffic is to
--/ cease
--/ the calling processor sends its elapsed time and epoch time to each
--/ processor on the network
--/ the calling processor waits until the full two-phase commit protocol
--/ (ready and received/implemented change) has been performed by all
--/ processors on the network, then unblocks network traffic
--/
--/ POSTCONDITIONS:
--/ all clocks are synchronized
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in a context that prohibits blocking
--/ (e.g., within an interrupt handler) and this call would block
--/ (i.e., all cases):
--/ the Kernel raises blocking_prohibited_exception
--/
--/ POSTCONDITIONS:
--/ the synchronize does not occur
--/ the exact state of network time is not known
--/ when a network problem is detected:
--/ the Kernel raises network_failure_exception
--/
--/ POSTCONDITIONS:
--/ the synchronize does not occur
--/ the exact state of network time is not known
--/ when more than one call to synchronize occurs:
--/ the Kernel raises synchronization_in_progress_exception

```

```

--/ POSTCONDITIONS:
--/ the synchronization is abandoned
--/ the exact state of network time is not known
--/

```

pragma (page);

```

procedure Synchronize (
  Timeout : In Time; Global Elapsed Time;
  Resumption_Priority : In Schedule.Types.Priority :=
    Schedule.Types.Current_Process_Priority);
--/ =====
--/ SUBPROGRAM NAME:
--/ synchronize
--/
--/ DESCRIPTION:
--/ forces all local processor clocks on kernel devices to synchronize
--/ time with the local clock on the invoking processor
--/
--/ --REQ: 12.1.16; 12.1.17; 12.1.18; 12.1.20; 12.1.21; 12.1.22; 12.1.27
--/
--/ --PRIM: 21.1.6
--/
--/ --TIME: 12.2.4; 12.2.5
--/
--/ PARAMETERS:
--/ timeout : after
--/ duration after which the process resends its attempt to
--/ synchronize processor clocks and becomes unblocked
--/
--/ default value:
--/ none
--/
--/ resumption_priority
--/ the priority to be assigned to this process when it becomes
--/ unblocked
--/
--/ default value:
--/ the calling process's current priority
--/
--/ PRECONDITIONS:
--/ there is no call to synchronize currently executing
--/
--/ ACTIONS:
--/ the calling processor indicates that all network traffic is to
--/ cease
--/
--/ the calling processor sends its elapsed time and epoch time to each
--/ processor on the network
--/
--/ the calling processor waits until the full two-phase commit protocol
--/ (ready and received/implemented change) has been performed by all
--/ processors on the network, then unblocks network traffic
--/
--/ POSTCONDITIONS:
--/ all clocks are synchronized
--/
--/ ERROR PROCESSING:
--/ when this primitive is called in a context that prohibits blocking
--/ (e.g., within an interrupt handler) and this call would block
--/ (i.e., all cases)
--/ the kernel raises blocking prohibited exception
--/
--/ POSTCONDITIONS:
--/ the synchronize does not occur

```

```

..  the exact state of network time is not known
..  when a network problem is detected:
..  the kernel raises network failure exception
..  POSTCONDITIONS:
..  the synchronize does not occur
..  the exact state of network time is not known
..  when more than one call to synchronize occurs:
..  the kernel raises synchronization in progress exception
..  POSTCONDITIONS:
..  the synchronization is abandoned
..  the exact state of network time is not known
..  when the synchronization timeout expires:
..  the kernel raises synchronization timed out exception
..  POSTCONDITIONS:
..  the synchronization is abandoned
..  the exact state of network time is not known
..
..

```

pragma (page);

```

procedure Synchronize (
  Timeout At : In Time Globals Epoch Time,
  Resumption Priority : In Schedule Types Priority :=
    Schedule Types Current Process Priority);
-- =====
-- SUBPROGRAM NAME:
-- synchronize
--
-- DESCRIPTION
-- forces all local processor clocks on kernel devices to synchronize
-- time with the local clock on the invoking processor
--
-- REQ: 12.1.16, 12.1.17, 12.1.18, 12.1.20, 12.1.21, 12.1.22, 12.1.27
--
-- PRIM: 21.1.6
--
-- TIME: 12.2.4, 12.2.5
--
-- PARAMETERS:
-- timeout at
-- absolute time at which the process rescinds its attempt to
-- synchronize processor clocks and becomes unblocked
--
-- default value:
-- none
--
-- resumption priority
-- the priority to be assigned to this process when it becomes
-- unblocked
--
-- default value:
-- the calling process's current priority
--
-- PRECONDITIONS:
-- there is no call to synchronize currently executing
--
-- ACTIONS
-- the calling processor indicates that all network traffic is to
-- cease
-- the calling processor sends its elapsed time and epoch time to each
-- processor on the network
-- the calling processor waits until the full two-phase commit protocol
-- (ready and received/implemented change) has been performed by all
-- processors on the network, then unblocks network traffic
--
-- POSTCONDITIONS
-- all clocks are synchronized
--
-- ERROR PROCESSING:
-- when this primitive is called in a context that prohibits blocking
-- (e.g., within an interrupt handler) and this call would block
-- (i.e., all cases):
-- the kernel raises blocking prohibited exception
-- POSTCONDITIONS
-- the synchronize does not occur

```



```

--/ the exact state of network time is not known
--/ when a network problem is detected:
--/ the Kernel raises network_failure exception
--/ POSTCONDITIONS:
--/ the synchronization does not occur
--/ the exact state of network time is not known
--/ when more than one call to synchronize occurs:
--/ the Kernel raises synchronization_in_progress exception
--/ POSTCONDITIONS:
--/ the synchronization is abandoned
--/ the exact state of network time is not known
--/ when the synchronization timeout expires:
--/ the Kernel raises synchronization_timeout exception
--/ POSTCONDITIONS:
--/ the synchronization is abandoned
--/ the exact state of network time is not known
--/
--/ =====

```

end Generic Time Management;
pragma (page);

```

--/ =====
--/ MODULE NAME:
--/ time management
--/
--/ MODULE TYPE:
--/ package specification
--/
--/ MODULE PURPOSE:
--/ an example of (and a template for) an instantiation of the generic
--/ package generic_time_management
--/
--/ =====
--/ MODULE DESCRIPTION:
--/ package instantiation the capabilities to define, modify, and read
--/ Kernel time where:
--/ error checking for change results in negative elapsed time is enabled
--/ error checking for change results in negative epoch time is enabled
--/ error checking for OK but time already passed is enabled
--/
--/ MODULE CONTENTS:
--/ time management (package specification)
--/ adjust elapsed time (procedure)
--/ adjust epoch time (procedure)
--/ change results in negative elapsed time (exception)
--/ change results in negative epoch time (exception)
--/ OK but time already passed (exception)
--/ read clock (function)
--/
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ none
--/
--/ USER'S MANUAL:
--/ Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/ none
--/
--/ NOTES:
--/ none
--/
--/ =====
--/ MODIFICATION HISTORY:
--/ 18aug88 bamborg created (complies clean)
--/
--/ =====
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/

```

```

../ Approved for public release, distribution is unlimited.
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628-85-C-0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the
../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../ .....
with Generic Time Management;
#pragma Elaborate (Generic Time Management);
.. to get: see MODULE CONTENTS
../ .....
package short name: TM
../ .....
package Time Management

Is new Generic Time Management {
Change Results In Negative Elapsed Time Enabled ==> True,
Change Results In Negative Epoch Time Enabled ==> True,
OK But Time Already Passed Enabled ==> True),
#pragma (page);

```

```

../ .....
../ MODULE NAME:
../ generic timeslice management
../
../ MODULE TYPE:
../ generic package specification
../
../ MODULE PURPOSE:
../ support timeslice scheduling of processes
../
../ .....
../ MODULE DESCRIPTION:
../ collection of subprograms to define the timeslice quantum and to
../ enable and disable timeslicing
../
../ MODULE CONTENTS:
../ generic timeslice management (generic package specification)
../ disable time slicing (procedure)
../ enable time slicing (procedure)
../ illegal quantum (exception)
../ illegal quantum enabled (generic formal parameter)
../ minimum slice time (constant)
../ minimum slice time value (generic formal parameter)
../ set timeslice (procedure)
../ default timeslice value (generic formal parameter)
../
../ REFERENCES:
../ DESIGN DOCUMENTS:
../ Kernel Facilities Definition
../ --REQ: 5.1.8
../ --REQ: 9.1.3; 9.1.4; 9.1.5; 9.1.10; 9.1.21; 9.1.24; 9.1.26
../ --PRIM: 18.1.6
../ --TIME: 9.2.4; 9.2.6; 9.2.7
../
../ USER'S MANUAL:
../ Kernel User's Manual
../
../ TESTING AND VALIDATION:
../ Unit testing
../
../ NOTES:
../ none
../
../ .....
../ MODIFICATION HISTORY
../ 18aug88 bamberg created (compiles clean)
../ 28sep88 dvk Changed to use package short names (to
../ conform to package body).
../ 13dec88 dvk Added generic parameter minimum_slice_time
../ 24mar88 rlv Bug report 0003 sel. Added ability to specify the initial default
../ timeslice
../
../ .....
../ VERSION: 3.0

```

```

with Kernel_Exceptions;
  .. to get: illegal_quantum

with Time_Globals;
  .. to get: elapsed_time, to_elapsed_time
  .. to get: kernel_time

generic
  .. ERROR checking conditional compilation flags
  .. REQ: 5.1.8

Illegal_Quantum_Enabled : In Boolean := True;

  .. minimum amount of time that can be specified as a timeslice interval

Minimum_Slice_Time_Value : In Time_Globals.Elapsed_Time;

  .. the default amount of time allocated to a process before it
  .. may be preempted
  .. default value:
  .. 1 second

Default_Timeslice_Value : Time_Globals.Elapsed_Time :=
  Time_Globals.To_Elapsed_Time(
    Kernel_Time_Kernel_Time'
    (High => 0, Low => 1_000_000));

  .. package short name: GTSM
  ..

package Generic_Timeslice_Management

is

package Tg

renames Time_Globals;
package Ke

```

```

with Kernel_Exceptions;
  .. to get: illegal_quantum

with Time_Globals;
  .. to get: elapsed_time, to_elapsed_time
  .. to get: kernel_time

generic
  .. ERROR checking conditional compilation flags
  .. REQ: 5.1.8

Illegal_Quantum_Enabled : In Boolean := True;

  .. minimum amount of time that can be specified as a timeslice interval

Minimum_Slice_Time_Value : In Time_Globals.Elapsed_Time;

  .. the default amount of time allocated to a process before it
  .. may be preempted
  .. default value:
  .. 1 second

Default_Timeslice_Value : Time_Globals.Elapsed_Time :=
  Time_Globals.To_Elapsed_Time(
    Kernel_Time_Kernel_Time'
    (High => 0, Low => 1_000_000));

  .. package short name: GTSM
  .. .....

package Generic_Timeslice_Management

is

package Tg

renames Time_Globals;
package Ke

```

```

renames Kernel_Exceptions;
--REQ: 9.1.3; 9.1.4; 9.1.5; 9.1.10; 9.1.21; 9.1.24; 9.1.26
--TIME: 9.2.4; 9.2.6; 9.2.7

-- illegal quantum raised by:
-- set_timeslice
-- PRIM: 18.1.6

Illegal_Quantum : exception renames Ke_Illegal_Quantum;
-- The minimum amount of time that can be specified as a timeslice interval

Minimum_Slice_Time : constant Tg_Elapsed_Time := Minimum_Slice_Time_Value;

pragma (page);

procedure set_timeslice (
  Quantum in Tg_Elapsed_Time);
--/
--/ SUBPROCESSOR NAME:
--/ set_timeslice
--/
--/ DESCRIPTION:
--/ set the timeslice quantum in elapsed time units
--/
--/ --REQ: 9.1.18
--/ --PRIM: 18.1.7
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ quantum
--/ the quantum of time allocated to a process for execution
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ the Kernel changes the length of the timeslice immediately and does
--/ not cause a reschedule point
--/
--/ POSTCONDITIONS:
--/ the length of the timeslice is adjusted as specified (which does not
--/ affect a previously pending timeslice for the currently running
--/ process)
--/
--/ ERROR PROCESSING:
--/ when a quantum is specified that is less than the value of
--/ minimum_slice_time:
--/ the Kernel raises illegal_quantum_exception
--/ POSTCONDITIONS:
--/ the requested change does not take place
--/ the length of the timeslice is set to the minimum_slice_time value
--/
--/
pragma (page);

```

```

procedure Enable Time Slicing;
--/ =====
--/ SUBPROGRAM NAME:
--/   enable_time_slicing
--/
--/ DESCRIPTION:
--/   enables round-robin, timeslice scheduling among processes of equal
--/   priority
--/
--/   --REQ: 9.1.19
--/   --PRM: 18.1.8
--/   --TIME: 9.2.5
--/
--/ PARAMETERS:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the Kernel enables time slicing immediately, which may later affect
--/   the currently running process, even though this is not a reschedule
--/   point
--/
--/ POSTCONDITIONS:
--/   time slicing is enabled for the processor
--/
--/ CAVEATS:
--/   It is possible to call enable_time_slicing without first calling
--/   set_timeslice. In this case, the default value specified for
--/   TSP.timeslice_duration is used as the timeslice amount. Note
--/   that this default value is not user configurable (due to a
--/   design error, and not due to any implementation restrictions).
--/   It is recommended that the first use of enable_time_slicing always
--/   be preceded by a call to set_timeslice.
--/
--/ ERROR PROCESSING:
--/   none
--/
--/ =====
pragma (page);

```

```

procedure Disable Time Slicing;
--/ =====
--/ SUBPROGRAM NAME:
--/   disable_time_slicing
--/
--/ DESCRIPTION:
--/   disables round-robin, timeslice scheduling among processes of equal
--/   priority
--/
--/   --REQ: 9.1.20
--/   --PRM: 18.1.9
--/   --TIME: 9.2.5
--/
--/ PARAMETERS:
--/   none
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   the Kernel disables time slicing immediately, which may affect the
--/   currently running process, even though this is not a reschedule
--/   point
--/
--/ POSTCONDITIONS:
--/   time slicing is disabled for the processor
--/
--/ ERROR PROCESSING:
--/   none
--/
--/ =====
end Generic Timeslice Management;
pragma (page);

```

```

./ .....
./ MODULE NAME
./ timeslice_management
./
./ MODULE TYPE
./ package specification
./
./ MODULE PURPOSE:
./ an example of (and a template for) an instantiation of the generic
./ package generic_timeslice_management
./
./ .....
./ MODULE DESCRIPTION:
./ package instantiating the capabilities to set a timeslice quantum and to
./ enable and disable time slicing where:
./ error checking for illegal_quantum is enabled
./
./ MODULE CONTENTS:
./ timeslice_management (package specification)
./ disable_time_slicing (procedure)
./ enable_time_slicing (procedure)
./ illegal_quantum (exception)
./ set_timeslice (procedure)
./
./ REFERENCES:
./ DESIGN DOCUMENTS:
./ none
./
./ USER'S MANUAL:
./ Kernel User's Manual
./
./ TESTING AND VALIDATION:
./ none
./
./ NOTES:
./ none
./
./ .....
./ MODIFICATION HISTORY:
./ 18aug88 bamberg created (compiles clean)
./ 13dec88 dvk Added generic parameter minimum_slice_time
./ 02aug89 dvk Added pragma elaborate to make VMS happy
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./
./ The following copyright must be included in this software and
./ all software utilizing this software
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.

```

```

./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85 C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....
with Time Globals;
.. to get: to elapsed time
with Kernel Time;
.. to get: kernel time;
with Generic Timeslice Management;
pragma Elaborate (Generic Timeslice Management);
.. to get: see MODULE CONTENTS
./ .....
./ package short name: TSM
./ .....
package Timeslice Management
is new Generic Timeslice Management (
illegal Quantum Enabled => True,
.. This value is set to 77 microseconds. Note that this is true
.. than the minimum context switch time (KFD : 2.6).
Minimum Slice Time Value => Time Globals.Ts Elapsed Time;
Kernel Time Kernel Time (H)H=>0, Low=>77);
pragma (page);

```

```

./ all software utilizing this software.
./
./ DISTRIBUTION:
./
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....

```

pragma (page):

```

./ .....
./ MODULE NAME
./ context_save_area
./
./ MODULE TYPE:
./ package specification
./
./ MODULE PURPOSE:
./ to encapsulate the hardware-specific information for saving the
./ context of a kernel process
./
./ .....
./ MODULE DESCRIPTION:
./ types and template for the hardware-specific information for saving the
./ context of a kernel process
./
./ .....
./ MODULE CONTENTS:
./ context_save_area (package specification)
./ address_registers_area (type)
./ context_save_area_contents (type)
./ data_registers_area (type)
./ floating_point_coprocessor_type (type)
./
./ REFERENCES:
./ DESIGN DOCUMENTS:
./ Kernel Facilities Definition
./ --REQ: 5.1.4
./ --PRIM: N/A
./ --TIME: N/A
./
./ USER'S MANUAL:
./ Kernel User's Manual
./
./ TESTING AND VALIDATION:
./ Integration testing with Scheduler
./
./ NOTES:
./ *** THIS IS HARDWARE SPECIFIC ***
./
./ *** NOTE THAT THE REP SPECS FOR ALL RECORDS ARE AT MOD 2 - WHICH LEAVES
./ *** STUFF AT 16-BIT BOUNDARIES ... THESE REALLY SHOULD BE A MOD 4 FOR
./ *** THE HARDWARE, BUT TELESOFT COMPILER DOESN'T SUPPORT THAT!!!
./
./ .....
./ MODIFICATION HISTORY:
./ 18aug88 bamberg created (complies clean)
./ 02sep88 bamberg added rep spec required by hardware instructions
./ 11jan89 firh final doc update for beta release (no change)
./
./ .....
./ VERSION: 3.0
./
./ DISCLAIMER:
./ The following copyright must be included in this software and

```

pragma (page);

```
.....
//package short name: CSA
//.....

with Hardware_Interface; use Hardware_Interface;
-- to get: bits per byte
--      hw long Integer
--      longword

package Context_Save_Area
is
-- *** THIS IS HARDWARE SPECIFIC ***
-- .. on the 68020, there are eight address registers, each 32-bits long
-- ..

type Address_Registers_Area is record
  A0 : Hw_Long_Integer;
  A1 : Hw_Long_Integer;
  A2 : Hw_Long_Integer;
  A3 : Hw_Long_Integer;
  A4 : Hw_Long_Integer;
  A5 : Hw_Long_Integer;
  A6 : Hw_Long_Integer;
  A7 : Hw_Long_Integer;
end record;

for Address_Registers_Area
--TS use record at mod hardware_interface longword;
--TS this is to get around a telesoft limitation: ideally, this should be
--TS allocated on a longword boundary, but telesoft does not support that
--TS via rep specs
use record at mod Hardware_Interface_Word;
  A0 at 0 * Hardware_Interface_Longword_range 0 .. 31;
  A1 at 1 * Hardware_Interface_Longword_range 0 .. 31;
  A2 at 2 * Hardware_Interface_Longword_range 0 .. 31;
  A3 at 3 * Hardware_Interface_Longword_range 0 .. 31;
  A4 at 4 * Hardware_Interface_Longword_range 0 .. 31;
  A5 at 5 * Hardware_Interface_Longword_range 0 .. 31;
  A6 at 6 * Hardware_Interface_Longword_range 0 .. 31;
  A7 at 7 * Hardware_Interface_Longword_range 0 .. 31;
end record;
-- this assumes a byte-addressable machine

Number_Address_Registers : constant := 8;

Address_Registers_Area_Size : constant :=
  Number_Address_Registers * Hardware_Interface_Bits_Per_Byte *
  Hardware_Interface_Longword;

for Address_Registers_Area_Size use Address_Registers_Area_Size;
```



```

... *** THIS IS HARDWARE SPECIFIC ***
...
... on the 68020, up to 280 bytes are used to save the state of the
... M68881 floating point coprocessor; when context is saved or restored,
... the address of this save area is passed to a hardware instruction that
... performs all the necessary manipulation for saving and restoring the
... M68881 floating point coprocessor

Number_Bytes_Floating_Point_Coprocessor_Save_Area : constant := 280;

type Floating_Point_Coprocessor_Type is array (
  1 .. Hw_Long_Integer /
    Number_Bytes_Floating_Point_Coprocessor_Save_Area /
    Hardware_Interface.Longword) of Hw_Long_Integer;
-- this assumes 4 bytes per word (hw_long_Integer)
-- this assumes that ONLY 280 bytes are allocated for this type;
-- if this proves to be false, it must be changed to a record type and
-- have rep specs written for it (as ada does not provide any rep specs
-- to assert sizes of array types)

Floating_Point_Coprocessor_Size : constant :=
  Number_Bytes_Floating_Point_Coprocessor_Save_Area *
  Hardware_Interface.Bits_Per_Byte;

for Floating_Point_Coprocessor_Type'Size use Floating_Point_Coprocessor_Size;

pragma (page);

```

```

... *** THIS IS HARDWARE SPECIFIC ***
...
... on the 68020, there are eight data registers, each 32 bits long

type Data_Registers_Area is record
  D0 : Hw_Long_Integer;
  D1 : Hw_Long_Integer;
  D2 : Hw_Long_Integer;
  D3 : Hw_Long_Integer;
  D4 : Hw_Long_Integer;
  D5 : Hw_Long_Integer;
  D6 : Hw_Long_Integer;
  D7 : Hw_Long_Integer;
end record;

for Data_Registers_Area
  use record at mod Hardware_Interface.Longword;
  -- TS use record at mod hardware interface longword;
  -- TS this is to get around a telesoft limitation: ideally, this should be
  -- TS allocated on a longword boundary, but telesoft does not support that
  -- TS via rep specs
  use record at mod Hardware_Interface.Word;
  D0 at 0 * Hardware_Interface.Longword range 0 .. 31;
  D1 at 1 * Hardware_Interface.Longword range 0 .. 31;
  D2 at 2 * Hardware_Interface.Longword range 0 .. 31;
  D3 at 3 * Hardware_Interface.Longword range 0 .. 31;
  D4 at 4 * Hardware_Interface.Longword range 0 .. 31;
  D5 at 5 * Hardware_Interface.Longword range 0 .. 31;
  D6 at 6 * Hardware_Interface.Longword range 0 .. 31;
  D7 at 7 * Hardware_Interface.Longword range 0 .. 31;
end record;
-- this assumes a byte-addressable machine

Number_Data_Registers : constant := 8;

Data_Registers_Area_Size : constant :=
  Number_Data_Registers * Hardware_Interface.Bits_Per_Byte *
  Hardware_Interface.Longword;

for Data_Registers_Area'Size use Data_Registers_Area_Size;

pragma (page);

```

```

... THIS IS HARDWARE SPECIFIC ...
.. the information required to be saved for a process context switch
.. Includes:
.. data registers
.. the contents of the 68020's eight data registers; this is used by
.. the Kernel's context switcher
.. default value:
.. none
..
.. address registers
.. the contents of the 68020's eight address registers; this is used
.. by the Kernel's context switcher
.. default value:
.. none
..
.. status register
.. the contents of the 68020's status register; this is used by the
.. Kernel's context switcher
.. default value:
.. none
..
.. program counter
.. the contents of the 68020's program counter; this is used by the
.. Kernel's context switcher
.. default value:
.. none
..
.. floating point coprocessor
.. the contents of the 68020's floating point coprocessor
.. default value:
.. none
..
type Context_Save_Area Contents is record
Data_Registers : Data_Registers_Area;
Address_Registers : Address_Registers_Area;
Status_Register : Hw_Long_Integer;
Program_Counter : Hw_Long_Integer;
Floating_Point_Coprocessor : Floating_Point_Coprocessor_Type;
end record;
.. for the data and address registers, these offset values are frozen into
.. the hardware movem operation, thus, this data structure must conform
.. with that
Data_Registers_Offset : constant := 0;

```

```

Address_Registers_Offset : constant := 8;
Status_Register_Offset : constant := 16;
Program_Counter_Offset : constant := 17;
Floating_Point_Coprocessor_Offset : constant := 18;

Status_Register_Size : constant :=
Hardware_Interface_Longword * Hardware_Interface_Bits_Per_Byte;
Program_Counter_Size : constant :=
Hardware_Interface_Longword * Hardware_Interface_Bits_Per_Byte;

for Context_Save_Area Contents
..TS use record at mod hardware_interface_longword;
..TS this is to get around a telesoft limitation: ideally, this should be
..TS allocated on a longword boundary, but telesoft does not support that
..TS via rep specs
use record at mod hardware_interface_word;
Data_Registers
at Data_Registers_Offset *
Hardware_Interface_Longword
range 0 .. Data_Registers_Area_Size - 1;
Address_Registers
at Address_Registers_Offset *
Hardware_Interface_Longword
range 0 .. Address_Registers_Area_Size - 1;
Status_Register
at Status_Register_Offset *
Hardware_Interface_Longword
range 0 .. Status_Register_Size - 1;
Program_Counter
at Program_Counter_Offset *
Hardware_Interface_Longword
range 0 .. Program_Counter_Size - 1;
Floating_Point_Coprocessor at Floating_Point_Coprocessor_Offset *
Hardware_Interface_Longword
range
0 .. Floating_Point_Coprocessor_Size - 1;
end record;

for Context_Save_Area Contents'Size use
Data_Registers_Area_Size +
Address_Registers_Area_Size +
Status_Register_Size +
Program_Counter_Size +
Floating_Point_Coprocessor_Size;
end Context_Save_Area;
pragma (page);

```

```

--/ .....
--/ MODULE NAME:
--/   generic_process_table
--/
--/ MODULE TYPE:
--/   generic_package_specification
--/
--/ MODULE PURPOSE:
--/   encapsulation of the process table data structure
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   defines and provides access to the process table data structure
--/
--/ .....
--/ MODULE CONTENTS:
--/   generic_process_table (generic package specification)
--/   acknowledged_message_information (type)
--/   adjacent_process_table_entry (type)
--/   communication_attributes_information
--/   create_process_information_record (function)
--/   destroy_process_information_record (procedure)
--/   event_identifier (subtype)
--/   event_information (type)
--/   initialize_process_table (procedure)
--/   locate_process (function)
--/   maximum_number_of_processes (constant)
--/   maximum_number_of_processes_value (generic formal parameter)
--/   null_event (constant)
--/   null_process (logical constant)
--/   null_semaphore (constant)
--/   pending_activity_attributes_information (type)
--/   process_attributes_information (type)
--/   process_identifier (type)
--/   process_information_record (type)
--/   process_initialization_status_type (type)
--/   process_table (generic package instantiation)
--/   process_table_entry (type)
--/   process_type (type)
--/   schedule_attributes_information (type)
--/   semaphore (type)
--/   semaphore_head (type)
--/   semaphore_head_ptr (type)
--/   semaphore_attributes_information (type)
--/   time_event_queue (generic package instantiation)
--/   tool_interface_attributes_information (type)
--/   "<=" (overloaded (2) function)
--/
--/ REFERENCES:
--/
--/ DESIGN DOCUMENTS:
--/   Kernel Facilities Definition
--/   --REQ: 5.1.6
--/   --REQ: 7.1.4, 7.1.6, 7.1.7
--/   --REQ: 7.1.9, 7.1.21, 7.1.22, 7.1.27
--/   --REQ: 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.1.5, 8.1.6,

```

```

--/   --REQ: 8.1.9, 8.1.10, 8.1.11, 8.1.16;
--/   --REQ: 9.1.7, 9.1.8, 9.1.9, 9.1.11, 9.1.12, 9.1.13;
--/   --REQ: 9.1.15, 9.1.16, 9.1.19, 9.1.20, 9.1.23, 9.1.29
--/   --REQ: 10.1.2, 10.1.3, 10.1.7, 10.1.8, 10.1.12;
--/   --REQ: 10.1.16, 10.1.17, 10.1.22, 10.1.37, 10.1.38, 10.1.39
--/   --REQ: 12.1.1, 12.1.13, 12.1.18, 12.1.19, 12.1.27
--/   --REQ: 13.1.2, 13.1.3, 13.1.7, 13.1.8, 13.1.9, 13.1.10, 13.1.11
--/   --PRIM: 16.1.1, 16.1.2
--/   --PRIM: 17.1.1
--/   --TIME: 8.2.1
--/
--/ Kernel Architecture Model
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Integration testing with rest of Kernel
--/
--/ NOTES:
--/   none
--/
--/ .....
--/ MODIFICATION HISTORY:
--/   18aug88 bamberg created (compiles clean, FINALLY!)
--/   22aug88 bamberg changes process_attributes_information names:
--/     local/remote process ID to local/remote index
--/     (these are integer indices, NOT process IDs)
--/   24aug88 bamberg added type process_type (kernel process or
--/     running on non-kernel device)
--/     added constants: null_process_index and
--/     null_hardware_device and used them
--/   25aug88 bamberg update to use generic_queue_manager and
--/     generic_storage_manager
--/     add ACK_event ID to pending_activity_attributes
--/     to break circular withs (datagram_globals)
--/   29aug88 bamberg move time event queue declaration into here
--/     real definitions for message_queue and
--/     current_send_buffer (datagram_pointer)
--/     correct comment on event_information_expiry
--/     move maximum_number_of_processes_value from
--/     GHM to here and export it as well
--/     establish common naming of processes as
--/     process_index; eliminate references to
--/     local/remote process indices
--/   31aug88 bamberg update to reflect changes in GQM
--/     (initial_allocation_gone; add_initialize_entry)
--/     add_procedure: initialize_process_table
--/   01sep88 bamberg incorporating new GQM changes (replace "<=" with
--/     "<=")
--/     clean up initializations
--/   02sep88 bamberg change type of maximum_message_queue_size to
--/     hw_long_positive (32 versus 16 bits); type of
--/     message_identifier to hw_long_integer (32 vs.
--/     16 bits)
--/   13sep88 rhs change pending_activities from booleans to a

```

```

../ single enumerated type, added acknowledged
../ message information record.
../ 14sep88 dvk moved message identifier to datagram_globals
../ change type of maximum message queue size to
../ hw_long natural (to allow zero queue length)
../ 14sep88 rvs changed null_process to a real null process,
../ and not just "null"
../ 17nov88 rvs added locate_process function...used to locate
../ a process by a brute force search of the process
../ table for the logical name
../ 12jan89 krth modified definition of semaphore and added
../ ancillary types semaphore_head( pb)
../ 15feb89 bamberg updated documentation on stack_high_address
../ to have example consistent with longword
../ alignment
../ 03apr89 rvs added next field to scheduling attributes
../ 03apr89 rvs added current_receive_buffer_address and
../ current_receive_buffer_size
../ fields to pending_activity_attributes
../ 12may89 rvs added real_time_interface structure
../ 16may89 rvs added size_of_process_table entry
../ 27aug89 bamberg final doc. check for 01sep89 freeze
../
../ .....
../ VERSION: 3.0
../
../ DISCLAIMER:
../
../ The following copyright must be included in this software and
../ all software utilizing this software.
../
../ DISTRIBUTION:
../
../ Approved for public release; distribution is unlimited
../
../ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
../ PA. The Software Engineering Institute (SEI) is a federally
../ funded research and development center established and operated
../ by Carnegie Mellon University (CMU). Sponsored by the U.S.
../ Department of Defense under contract F19628 85 C 0003, the SEI
../ is supported by the services and defense agencies, with the U.S.
../ Air Force as the executive contracting agent.
../
../ Permission to use, copy, modify, or distribute this software and
../ its documentation for any purpose and without fee is hereby
../ granted, provided that the above copyright notice appear in all
../ copies and that both that copyright notice and this permission
../ notice appear in supporting documentation. Further, the names
../ of Software Engineering Institute or Carnegie Mellon University may
../ not be used in advertising or publicity pertaining to
../ distribution of the software without specific, written prior
../ permission. CMU makes no claims or representations about the
../ suitability of this software for any purpose. This software is
../ provided "as is" and no warranty, express or implied, is made
../ by the SEI or CMU, as to the accuracy and functioning of the

```

```

../ program and related program material, nor shall the fact of
../ distribution constitute any such warranty. No responsibility is
../ assumed by the SEI or CMU in connection herewith.
../ .....
../ pragma (page);

```

```

with Communication_Globals;
.. to get: message_length_type

with Context_Save_Area;
.. to get: context_save_area_contents

with Context_Switcher_Globals;
.. to get: context_saved_type

with Datagram_Globals;
.. to get: datagram_pointer
.. message_identifier

with Generic_Queue_Manager;
.. to get: all_utilities

with Hardware_Interface; use Hardware_Interface;
.. to get: hw_address
.. hw_integer
.. hw_long_natural
.. hw_natural
.. hw_positive
.. hw_string

with Kernel_Time;
.. to get: Kernel_time
.. zero_kernel_time

with Kernel_Exceptions;
.. to get: Kernel_exceptions

with Network_Globals;
.. to get: process_index_type

with Process_Managers_Globals;
.. to get: how_to_handle_message_queue_overflow
.. maximum_length_of_process_name

with Schedule_Types;
.. to get: preemption
.. priority
.. process_state

with Time_Keeper_Globals;
.. to get: event_type
.. time_class_type

with Tool_Interface_Globals;
.. to get: collection_type,

pragma (page);

```

```

generic
..
.. value indicating the maximum number of processes in the Process Table
.. (this value is never used within the Kernel except for the Initialization
.. of the size of the Process Table, nor should it be used without the
.. Kernel)
..
.. this value indicates the maximum size of the Process Table as determined
.. when initialization is complete, not during the initialization process
..
.. REQ: 5.1.6
..
Maximum_Number_Of_Processes_Value : In_Hw_Positive;

package Generic_Process_Table
is
.. short name: GPTB
..
.. value indicating the maximum number of processes in the Process Table
.. this value indicates the maximum size of the Process Table as determined
.. when initialization is complete, not during the initialization process
..
.. REQ: 5.1.6
..
Maximum_Number_Of_Processes : constant Hw_Positive :=
Maximum_Number_Of_Processes_Value;
..
.. a process identifier is used to access process table entries efficiently
..
type Process_Information_Record;

type Process_Identifier is access Process_Information_Record;
..
.. a null process identifier is just a pointer to an empty process (this
.. is set in initialize process table)
..
Null_Process : Process_Identifier;
pragma (page);

```

```

-- .. event identifier is used by the Kernel's time keeper to access
-- .. time event queue entries efficiently
-- ..
-- .. event information is the template used to represent each of the sequence
-- .. of events that are understood by the Kernel to happen some time in the
-- .. future (i.e., the template for each entry in the time event queue)
-- ..
-- .. the information maintained for each time event is:
-- ..   kind of event
-- ..   the event type causing the creation of this time event
-- ..
-- .. possible values:
-- ..   alarm (via a call to set_alarm)
-- ..   receive timeout (via a call to receive_message)
-- ..   semaphore timeout (via a call to claim)
-- ..   wait timeout (via a call to wait)
-- ..   send with ACK timeout (by receiving a message indicating a
-- ..     remote timeout is required)
-- ..   slice expiration (time slicing was previously enabled)
-- ..
-- .. default value:
-- ..   none
-- ..
-- .. this value is assigned when an event information record is added
-- .. to the time event queue and should never change as long as it is
-- .. still in the time event queue
-- ..
-- .. time class
-- .. the type of time specified by the application (via a call to any
-- .. Kernel primitive that provides a time parameter)
-- ..
-- .. possible values:
-- ..   elapsed (the application specified an elapsed time)
-- ..   epoch (the application specified an epoch_time)
-- ..
-- .. default value:
-- ..   none
-- ..
-- .. this value is assigned when an event information record is added
-- .. to the time event queue and should never change as long as it is
-- .. still in the time event queue
-- ..
-- .. expiry
-- .. the kernel time at which the time event expires; the time in each
-- .. entry is that which was specified via the Kernel primitive called
-- .. default value:
-- ..   none
-- ..
-- .. this value is assigned when an event information record is added

```

```

function "<=" (Left_Event_Information)
return Boolean;
--=====
--/ SUBPROGRAM NAME:
--/ "<="
--/
--/ DESCRIPTION:
--/ defines the "less than or equal to" operation on two
--/ event information records in terms of comparing their .expiry
--/ components
--/ --REQ: 5.1.4
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ left
--/ the left operand of the comparison between two event_information
--/ records
--/ default value:
--/ none
--/ right
--/ the right operand of the comparison between two event_information
--/ records
--/ default value:
--/ none
--/ function return value
--/ result of the comparison
--/ values include
--/ true (the left operand orders "less than or equal to" the right)
--/ false (the left operand does not order "less than or equal to"
--/ the right)
--/ PRECONDITIONS:
--/ none
--/ ACTIONS:
--/ return true if left.expiry <= right.expiry
--/ POSTCONDITIONS:
--/ none
--/ ERROR PROCESSING:
--/ none
--/
--/ =====
pragma (page);

```

.. /
..

.. a semaphore is defined by the Kernel; it provides the
.. capability to claim and release it - that is all

.. REQ: 8.1.1; 8.1.2
.. PRIM: 17; 17.1.1

.. To allow a semaphore to function reliably, two things
.. must be guaranteed

.. (a) operations upon semaphores must use sharing semantics, i.e.
.. to operations applied concurrently to the same semaphore
.. must access the exact same data structure

.. (b) all semaphores must be initialized upon declaration

.. To ensure (a), the implementation allocates all semaphore objects
.. and references them by access values

.. To ensure (b), all types associated with semaphores are record types
.. with default initialization of components

.. The three types that are declared below are

.. Semaphore_Head
.. the type of a semaphore object

.. Semaphore_Head_Ptr
.. an access type designating a semaphore_head

.. Semaphore
.. a record type whose only component is a semaphore_head_pointer
.. with appropriate initialization

.. The last type - Semaphore - is the only one that the primitives
.. operate on, and the only one of interest to the application

.. the information maintained each semaphore head is:
.. number of waiting processes
.. the number of processes in the waiting queue for the semaphore

.. default value:

.. -1 (there are no processes waiting and the semaphore is free;
.. the value 0 indicates that the semaphore is claimed and
.. there are no processes waiting for it; any positive value
.. indicates that the semaphore is currently claimed and there
.. are positive value number of processes waiting for the
.. semaphore)

.. this value is incremented via a call to the Kernel primitive:
.. claim and decremented via a call to the Kernel primitive: release;

.. otherwise, this value should never change
..
.. queue_head
.. the first process waiting in the queue for this semaphore
..
.. default value:
.. null_process (there are no processes waiting in the queue for
.. this semaphore)
..
.. this value may be set via a call to the Kernel primitive: claim
.. (if the call is the first process waiting for the semaphore)
.. and reset via a call to the Kernel primitive: release (if the call
.. is for the last process waiting for the semaphore); otherwise,
.. this value should never change
..
.. sema_previously_claimed
.. the last semaphore claimed by the process that owns this semaphore
.. (i.e. if process P claims semas S1 and S2 in that order, then this
.. component of S2 head will designate S1)

.. default value: null_semaphore

.. The purpose of this component is to chain together in LIFO order
.. all semas currently owned by a process.

.. REQ: 8.1.1; 8.1.2

.. PRIM: 17.1.1

.. TIME: 8.2.1

.. preliminary incomplete type declaration

type Semaphore_Head;

.. the access type, to allow sharing semantics on semaphore objects

type Semaphore_Head_Ptr is access Semaphore_Head;

.. the type of a semaphore object proper, with correct initialization

type Semaphore_Head is record

Number_Of_Waiting_Processes : hw_Integer := -1;

Queue_Head : Process_Identifier := Null_Process;

Sema_Previously_Claimed : Semaphore_Head_Ptr := null;

end record;

.. the type visible to the application, with correct initialization
.. each declaration thus creates a new semaphore object, which will
.. be automatically initialized.

type Semaphore is record

Head : Semaphore_Head_Ptr := new Semaphore_Head;

end record;

.. the null semaphore value


```
-- this must be the initial value of all semaphore variables
-- that are 'working' variables, ie that are not associated with
-- unique semaphore objects but rather that are assigned semaphore
-- values dynamically
```

```
Null Semaphore : constant Semaphore := (Head => null );
pragma (page);
```

```
./ .....
--
-- Indication of current status of process initialization protocol
-- (i.e., process declaration and creation)
-- components include:
--   declared
--   Indication of whether or not declare_process successfully completed
--
-- values include:
--   true (successful completion of declare_process)
--   false (unsuccessful completion of declare_process)
--
-- default value:
--   false (unsuccessful completion of declare_process)
--
-- this value is set via a call to the Kernel primitive:
--   declare_process; it should never change after this call
--
-- created
--   Indication of whether or not create_process successfully completed
--
-- values include:
--   true (successful completion of create_process)
--   false (unsuccessful completion of create_process)
--
-- default value:
--   false (unsuccessful completion of create_process)
--
-- this value is set via a call to the Kernel primitive:
--   create_process; it should never change after this call
--
-- remotely_created
--   Indication of whether or not the process was created on another
--   node
--
-- values included:
--   true (process was created on a remote node)
--   false (process was not created remotely)
--
-- default value:
--   false (no knowledge yet about where the process was created)
--
-- This value is set by the receive_datagram_interrupt_handler when
-- ever a process_created message arrives.
--
-- REQ: 7.1.4
-- PRIM: 16.1.1, 16.1.2
--
type Process Initialization_Status_Type is record
  Declared : Boolean := False;
  Created : Boolean := False;
  Remotely_Created : Boolean := False;
end record;
```

```

.. .. Indication of whether or not a process is running on a Kernel processor
.. .. (and thus follows all Kernel protocols) or a process is really just a
.. .. non-Kernel device (and thus follows none of the Kernel protocols)
.. ..
.. .. values include:
.. ..   Kernel_process (follows Kernel protocols)
.. ..   non_Kernel_device (does not follow Kernel protocols)
.. ..
.. .. PRIM: 16.1.1

```

```

type Process_Type is (
  Kernel_Process, Non_Kernel_Device);
pragma (page);

```

```

.. .. process attributes include:
.. ..   logical_name
.. ..     the string valued name provided by the application for a process;
.. ..     the length of the string is limited by the user-provided value
.. ..     for the maximum length of a process name
.. ..
.. ..   default_value:
.. ..     none
.. ..
.. ..   this_value_is_set_via_a_call_to_the_Kernel_primitive:
.. ..     declare_process; it should never change after that call
.. ..
.. ..   kind_of_process
.. ..     indication of whether a process is running on a Kernel processor
.. ..     (and thus follows all Kernel protocols) or a process is really
.. ..     just a non-Kernel device (and thus follows none of the Kernel
.. ..     protocols)
.. ..
.. ..   values include:
.. ..     Kernel_process (follows Kernel protocols)
.. ..     non_Kernel_device (does not follow Kernel protocols)
.. ..
.. ..   default_value:
.. ..     Kernel_process
.. ..
.. ..   this_value_is_set_via_a_call_to_the_Kernel_primitive:
.. ..     declare_process; it should never change after that call
.. ..
.. ..   process_initialization_status
.. ..     indication of current status of process initialization protocol
.. ..     (i.e., process declaration and creation)
.. ..
.. ..   default_value:
.. ..     declared => false (set by declare_process)
.. ..     created => false (set by create_process)
.. ..
.. ..   this_value_is_set_via_a_call_to_the_Kernel_primitives:
.. ..     declare_process (the declared component) and create_process (the
.. ..     created component); it should never change after the calls
.. ..
.. ..   process_index
.. ..     another way of referencing a process (in addition to a process
.. ..     identifier) via its owning processor and an identifier unique with
.. ..     respect to that processor
.. ..
.. ..   default_value:
.. ..     none
.. ..
.. ..   for_a_non_Kernel_device_this_value_is_set_via_a_call_to_the_Kernel
.. ..   primitive: declare_process; it should never change after that call;
.. ..   for_a_Kernel_device_this_value_is_set_via_a_call_to_the_Kernel
.. ..   primitive: create_process; it should never change after that call
.. ..
.. ..   code_address

```

```

..
..
.. the address of the code that comprises this process
..
.. default value:
.. none
..
.. this value is set via a parameter to the Kernel primitive:
.. create_process; it should never change after that call
..
..
.. stack_low_address
.. the system low address (e.g., 16#0000) of the Kernel-created
.. process stack; this is the FIRST longword address (i.e., aligned
.. on a 32-bit boundary) at which the Kernel may safely store a
.. longword (i.e., 32 bits) of data in the Kernel-maintained
.. process stack
..
.. *** the Kernel-maintained process stack is always longword-aligned
..
..
.. default value:
.. none
..
.. this value is set via a call to the Kernel primitive:
.. create_process; it should never change after that call
..
..
.. stack_high_address
.. the system high address (e.g., 16#FFC0) of the Kernel-created
.. process stack; this is the LAST longword address (i.e., aligned
.. on a 32-bit boundary) at which the Kernel may safely store a
.. longword (i.e., 32 bits) of data in the Kernel-maintained
.. process stack
..
.. *** the Kernel-maintained process stack is always longword-aligned
..
..
.. default value:
.. none
..
.. this value is set via a call to the Kernel primitive:
.. create_process; it should never change after that call
..
..
.. context_saved
.. Indication of whether or not the current context of this process
.. may be assumed to be saved (i.e., that the context_save_area has
.. contents that are currently valid)
..
..
.. values include:
.. via_call (context is saved and was saved via the
.. procedure/function calling protocol)
.. via_interrupt (context is saved as was saved via the
.. interrupt handling protocol)
.. not_saved (context must not be assumed to be saved)
..
.. default value:
.. via_call
..
..
.. this value is set initially via a call to the Kernel primitive:
.. create_process; it is modified by the Kernel as process context is

```

```

..
..
.. saved and restored (when a process context switch occurs as
.. directed by the Scheduler or when an interrupt occurs)
..
..
.. process_context_save_area
.. place where the context of a process is saved (e.g., registers,
.. program counter)
..
..
.. default value:
.. none
..
..
.. this value is modified by the Kernel as process context is
.. saved and restored (when a process context switch occurs as
.. directed by the Scheduler or when an interrupt occurs)
..
.. --REQ: 7.1.4; 7.1.6; 7.1.7
..
..
.. Type Process Attributes Information is record
.. Logical Name: Hw String (1 ..
.. Positive (Process_Managers_Globals.Maximum_Length_Of_Process_Name)) :=
.. (others => '');
.. Kind_Of_Process: Process_Type := Kernel_Process;
.. Process_Initialization_Status: Process_Initialization_Status_Type;
.. Process_Index: Network_Globals.Process_Index_Type;
.. Code_Address: Hw_Address;
.. Stack_Low_Address: Hw_Address;
.. Stack_High_Address: Hw_Address;
.. Context_Saved: Context_Switcher_Globals.Context_Saved_Type :=
.. Context_Switcher_Globals.Via_Call;
.. Process_Context_Save_Area: Context_Save_Area.Context_Save_Area_Contents;
.. end record;
..
.. pragma (page);

```

```

... schedule attributes include:
... state
... the current state of this process; this is used by the Kernel's
... Scheduler
...
... values include:
... running (this process controls the processor and is the
... currently running process)
... suspended (this process is able to run but another process is
... currently running)
... blocked (this process is unable to run)
... dead (this process is no longer able to run)
...
... default value:
... blocked (this process is unable to run, since it has not yet
... been inserted into the scheduler)
...
... this value is set by the Scheduler as the process state changes
... (due to a call to a Kernel primitive, the passage of time, the
... occurrence of an event)
...
... priority
... the current priority of this process; this is used by the Kernel's
... Scheduler, and the primitive.
... process attribute readers get process priority, and all primitives
... take a (resumption) priority as a parameter
...
... default value:
... none
...
... this value is set initially via a parameter to the Kernel primitive:
... create process; it may be modified by the Scheduler as the process
... priority changes
... (due to a call to a Kernel primitive, the passage of time, the
... occurrence of an event)
...
... preemption
... an indication of whether or not this process may be preempted; this
... is used by the Kernel's Scheduler (for time slicing), and the
... primitives.
... process attribute readers get process preemption
... process attribute modifiers set process_preemption
...
... values include:
... true (this process may be preempted)
... false (this process may not be preempted)
...
... default value
... none, provided by initial call to create process
...
... this value is set initially via a parameter to the Kernel primitive:
... create process; it may be modified via a call to the Kernel
... primitive: set process_preemption

```

```

... next
... a pointer to the next process in the scheduler's run queue
...
... values include:
... any legal process identifier for a Kernel process
...
... default value:
... null
...
... REQ: 9.1.7; 9.1.8; 9.1.9; 9.1.11; 9.1.12; 9.1.13; 9.1.23
...
... type Schedule Attributes Information is record
... State : Schedule_Type.Process_State := Schedule_Type.Blocked;
... Priority : Schedule_Type.Priority;
... Preemption : Schedule_Type.Preemption;
... Next_Process_Identifier : null;
... end record;
... pragma (page);

```

```

.. communication attributes include:
.. next available message ID
.. the message ID that may be used for the next message sent by this
.. process via send_message_and_wait; this value is constantly
.. increasing
.. default value
.. first (lowest) message identifier available
.. this value is modified only via a call to the Kernel primitive:
.. send_message_and_wait
.. maximum message_queue_size
.. the maximum number of messages that may be queued awaiting receipt
.. for this process
.. default value:
.. none
.. this value is set by a parameter to the Kernel primitive:
.. create_process; it should never be modified after that call
.. message_queue
.. pointer to the first message in the message queue for this process;
.. this is used by the Kernel primitives to send and receive messages
.. default value:
.. null
.. this value is set via a call to the Kernel primitive:
.. create_process; it should never be modified after that call
.. current_send_buffer
.. pointer to the current buffer being used to send a message; via
.. an application call to send_message*
.. default value:
.. null
.. this value is set via a call to the Kernel primitives:
.. send_message and send_message_and_wait; it is also used by
.. GIPAM purge_message_queue to clean out the message of a
.. recipient of a "kill" message
.. queue_overwrite_rule
.. indication of how this process is to handle incoming message queue
.. overflow
.. values include
.. drop_newest_message (the most recently received message is lost)
.. default value:
.. none
..
.. this value is set by a parameter to the Kernel primitive:
.. create_process; it should never change after that call
.. message_queue_overflow
.. indication of whether or not the incoming message queue for this
.. process is currently full and messages are being lost or in danger
.. of being lost; this is used by the Kernel primitive:
.. receive_message
.. values include:
.. true (at least one message has been lost already)
.. false (no messages have been lost since last call to
.. receive_message)
.. default value:
.. false (no messages have been lost since last call to
.. receive_message)
.. this value may be set by the Kernel as messages are received; its
.. value may be reset via a call to the Kernel primitive:
.. receive_message
..
.. REQ: 7.1.9, 7.1.21, (7.1.22), 7.1.27
.. REQ: 10.1.2, 10.1.3, 10.1.12, 10.1.22
..
type Communication_Attributes Information is record
Next_Available_Message_Id : Datagram_Globals.Message_Identifier := 0;
Maximum_Message_Queue_Size : Hw_Long_Natural;
Message_Queue : Datagram_Globals.Datagram_Pointer := null;
Current_Send_Buffer : Datagram_Globals.Datagram_Pointer := null;
Queue_Overwrite_Rule :
Process_Managers_Globals.How_To_Handle_Message_Queue_Overflow;
Message_Queue_Overflow : Boolean := False;
end record;
pragma (page);

```

```

... the kinds of mutually exclusive activities that can be pending for a
... single process are
... receive_pending (the application called the Kernel primitive:
...   receive_message and is blocked until a message is received or until
...   the timeout expires)
... semaphore_pending (the application called the Kernel primitive:
...   claim and is blocked until the requested semaphore is free or until
...   the timeout expires)
... send_with_ack_pending (the application called the Kernel primitive:
...   send_message and wait and is blocked until an ACK or a NAK is
...   returned or until the timeout expires)
... wait_pending (the application called the Kernel primitive: wait and is
...   blocked until the timeout expires)
... nothing_pending (there is no activity on which this process is
...   currently pending)

```

```

type Pending_Activity_Type is (
  Receive_Pending,
  Semaphore_Pending,
  Send_With_Ack_Pending,
  Wait_Pending,
  Nothing_Pending);

```

pragma (page);

```

... pending activity attributes includes:
... pending_activity
...   Indication of what kind of event has caused the process to block
... values include:
...   see pending_activity_type just above
... default value:
...   nothing_pending
... this value is set via a call to any blocking Kernel primitive:
...   receive_message, claim, send_message and wait; it is
...   reset by the Kernel upon expiry of the timeout, occurrence of
...   the event awaited (e.g., receipt of message or ACK/NAK, availability
...   of the semaphore)
... pending_event_ID
...   an index into the time_keeper's time event queue indicating the
...   event entry corresponding to the value of pending_activity for this
...   process;
...   used by Kernel Internals as a link into the Kernel's time_keeper
... default value:
...   null_event
... this value is set via a call to any blocking Kernel primitive (as
...   enumerated above); it is reset by the Kernel as described above
... current_pending_message
...   if pending_activity indicates send_with_ack_pending, this is the
...   message identifier for which an ACK or a NAK is expected
... default value:
...   none
... this value is set via a call to the Kernel primitive:
...   send_message and wait; it is reset by the Kernel upon receipt of
...   the ACK/NAK for the identified message; it is valid if and only
...   if pending_activity indicates send_with_ack_pending
... current_receive_buffer_address
...   address of the receiver's message buffer. Used to perform local
...   optimizations.
... default value:
...   none
... current_receive_buffer_size
...   size of the data space accessed via current_receive_buffer_address.
...   Used to perform local optimizations.
... default value:
...   none

```

```

alarm_event_id
  an index into the time keeper's time event queue indicating the
  alarm expiration event for this process

default_value:
  null event

  this value is set via a call to the Kernel primitive: set_alarm;
  it may be reset either via a call to the Kernel primitive:
  cancel_alarm or by the Kernel upon the expiry of the alarm

alarm_resumption_priority
  if alarm_event_id is not the null event, the priority at which
  this process is to be resumed upon the expiration of the alarm

default_value:
  none

  this value is set via a call to the Kernel primitive: set_alarm;
  it should never change otherwise; it is valid if and only if
  alarm_event_id is not the null_event

exception_name
  indication of whether or not the Kernel is raising an exception
  for this process; if not no_exception, then also an indication of
  which exception is to be raised

values include:
  there is an enumeration literal corresponding to each exception
  the Kernel may raise; see package Kernel_exceptions

default_value:
  no_exception

  this value is set whenever the Kernel Internals detect a Kernel
  exception that is to be raised and reset to no_exception upon
  completion of internal exception processing

--REQ: 8.1.5; 8.1.6; 8.1.15; 9.1.16; 9.1.19; 9.1.20; 9.1.29
--REQ: 10.1.7; 10.1.8; 10.1.16; 10.1.17; 10.1.37; 10.1.38; 10.1.39
--REQ: 12.1.11; 12.1.13; 12.1.18; 12.1.19; 12.1.27
--REQ: 13.1.2; 13.1.3; 13.1.7; 13.1.8; 13.1.9; 13.1.10; 13.1.11

```

```

type Pending_Activity_Attributes_Information is record
  Pending_Activity : Pending_Activity_Type := Nothing Pending;
  Pending_Event_Id : Event_Identifier := Null_Event;
  Current_Pending_Message : Datagram_Globals_Message_Identifier;
  Current_Receive_Buffer_Size : Communication_Globals_Message_Length_Type;
  Current_Receive_Buffer_Address : Hw_Address;
  Alarm_Event_Id : Event_Identifier := Null_Event;
  Alarm_Resumption_Priority : Schedule_Types_Priority;
  Exception_Name : Kernel_Exceptions.Kernel_Exceptions :=
    Kernel_Exceptions.No_Exception;
end record;

```

pragms (page):

```

... acknowledged message information is maintained in the process table
... entry corresponding to the SENDING process; this information refers to
... data about the RECEIVING process's incoming message queue, this is done
... to facilitate ready access to message queue information to process
... timeout expiration efficiently
...
... acknowledged message information includes:
... event ID
... indication that this process is the SENDING process and
... sent a message via the Kernel
... primitive: send_message and wait; thus an ACK is required to be
... returned to THIS process upon receipt of the corresponding message
...
... default value:
... null event
...
... this value is set by the Kernel when it receives a message that
... was sent via a call to the Kernel primitive: send_message and wait;
... it is reset via a call to the Kernel primitive: receive_message
... or by the Kernel when the corresponding timeout expires
...
... message
... an index into the RECEIVING process's incoming message queue
... indicating the message that this process sent via the Kernel
... primitive: send_message and wait
...
... default value:
... null
...
... this value is set by the Kernel when it receives a message that
... was sent via a call to the Kernel primitive: send_message and wait;
... it is reset via a call to the Kernel primitive: receive_message
... or by the Kernel when the corresponding timeout expires
...
... queue
... a pointer to the head of the RECEIVING process's message queue -
... i.e., the message queue that contains the message field just above
...
... default value:
... null
...
... this value is set by the Kernel when it receives a message that
... was sent via a call to the Kernel primitive: send_message and wait;
... it is reset via a call to the Kernel primitive: receive_message
... or by the Kernel when the corresponding timeout expires
...
... REQ 10.1.3

```

```

type Acknowledged Message Information is record
Event ID : Event Identifier := Null Event;
Message : Datagram; Globals Datagram Pointer := null;
Queue : Datagram; Globals Datagram Pointer := null;
end record;

```

pragma (page);


```

-- semaphore attributes include:
--
-- sema last claimed
-- the identity of the semaphore most recently claimed by
-- the process, and still owned by it.
--
-- default value
-- null semaphore
--
-- semaphore name
-- the identity of the semaphore on which this process is currently
-- waiting
--
-- default value:
-- null semaphore
--
-- this value is set via a call to the Kernel primitive: claim;
-- it is reset by release
--
-- next process pending claim
-- the process identifier for the process that called the Kernel
-- primitive claim after this process did
--
-- default value:
-- null process ID
--
-- this value is set by the next call of the Kernel primitive: claim;
-- it may be reset by the Kernel if the timeout of that claim expires
--
-- previous process pending claim
-- the process identifier for the process that called the Kernel
-- primitive claim before this process did
--
-- default value:
-- null process ID
--
-- this value is set via the current call to the Kernel primitive:
-- claim; it is reset by the Kernel if the timeout of the claim
-- expires or once the previous process releases the semaphore
--
-- REQ. 8.1.3; 8.1.4; 8.1.9; 8.1.10; 8.1.11
--
-- type Semaphore Attributes Information is record
--
-- Sema Last Claimed : Semaphore := Null Semaphore;
-- Semaphore Name : Semaphore := Null Semaphore;
-- Next Process Pending Claim : Process Identifier := Null Process;
-- Previous Process Pending Claim : Process Identifier := Null Process;
-- end record;
--
-- pragma (page);

```

```

-- These structures support the collection of data related to process
-- activities, i.e., they support the tool interface into the Kernel.
--
-- monitoring process type
-- This array holds process identifiers. One process identifier
-- for each possible attribute that can be monitored, this allows
-- different attributes to be sent to different tool processes.
--
-- default value:
-- none
--
-- monitoring message tag type
-- this array holds the message tags. One message tag for each
-- possible attribute that can be monitored (this is tag that the
-- tool process will receive when a message is sent about the
-- attribute). This allows each tool interface message to be
-- tagged in a manner defined by the application.
--
-- default value:
-- none
--
-- tool interface attributes information
-- this record contains all the information the tool process
-- needs to collect data on other processes in the system.
--
-- enabled
-- this variable indicates whether or not some tool process
-- is monitoring the activity of this particular process.
--
-- default value:
-- false
--
-- this value is set to true when begin collection is called
-- with this process's pid and set to false when
-- cease collection is for this process and ALL its
-- collection attributes are disabled.
--
-- monitoring process
-- this array holds the pid of the tool process that will
-- receive the index attribute message.
--
-- default value:
-- null
--
-- this value is set by a call to begin collection and
-- reset to null by a call to cease collection
--
-- message tag
-- this array holds the tags for each tool interface message
-- generated for the index attribute for this process
--
-- default value:
-- null;

```

```

.. this value is set by a call to begin_collection
..
type Monitoring_Process_Type is array
  (Tool_Interface_Globals.Collection_Type) of Process_Identifier;

type Monitoring_Message_Tag_Type is array
  (Tool_Interface_Globals.Collection_Type) of
    Communication_Globals.Message_Tag_Type;

type Tool_Interface_Attributes_Information is record
  Enabled: Boolean := False;
  Monitoring_Process_Type := (others => null);
  Message_Tag: Monitoring_Message_Tag_Type;
end record;

pragma page;

.. each process information record comprises the following:
.. process_attributes
.. the collection of process attribute information defined above
..
.. schedule_attributes
.. the collection of schedule attribute information defined above
..
.. communication_attributes
.. the collection of communication attribute information defined above
..
.. pending_activity_attributes
.. the collection of pending activity attribute information defined
.. above
..
.. send_w_ack_attributes
.. the collection of attributes relating to the sending process of
.. a message sent via the Kernel primitive: send_message_and_wait
..
.. semaphore_attributes
.. the collection of semaphore attribute information defined above
..
.. tool_interface_attributes
.. the collection of tool interface attribute information defined above
..
..
type Process_Information_Record is record
  Process_Attributes: Process_Attributes_Information;
  Schedule_Attributes: Schedule_Attributes_Information;
  Communication_Attributes: Communication_Attributes_Information;
  Pending_Activity_Attributes: Pending_Activity_Attributes_Information;
  Send_W_Ack_Attributes: Acknowledged_Message_Information;
  Semaphore_Attributes: Semaphore_Attributes_Information;
  Tool_Interface_Attributes: Tool_Interface_Attributes_Information;
end record;

.. the process table is a collection of process information records, built
.. dynamically; the process identifier points to a
.. process information record, which was allocated and initialized during
.. process initialization time (e.g., via declare_process_and_create_process
.. calls)
..
.. the list of process table entries contains a processor identifier and
.. is maintained by an instantiation of the generic_queue_manager
..
..
.. entries in the process table comprise:
.. process_ID
.. this is the real reference to the information specific to this
.. process; this is the value, cast as a
.. process_type.process_identifier, that the application uses
.. when referencing a process anywhere in the application program

```

```

-- default : 32.2.
-- none
--
-- this value is set via a call to the Kernel primitive
-- declare process, this value should never change after the call
--
-- next and previous pointers, which are maintained by the instantiation
-- of the generic queue manager
--
type Process_Table_Entry is record
  Process_Id : Process_Identifier := Null_Process;
end record;

type Adjacent_Process_Table_Entry is access Process_Table_Entry;
pragma (page);

```

```

function "<=" (Left, Right : Process_Table_Entry)
return Boolean;
--=====
--/ SUBPROGRAM NAME:
--/ "<="
--/
--/ DESCRIPTION:
--/ defines a less-than-or-equal-to ordering for process table entries,
--/ which always returns false (i.e., new entries will always be inserted
--/ at the head of the list)
--/
--/ --REQ: 5.1.4
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the comparison between process table entries
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the comparison between process table entries
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the comparison; always false
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ return true
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/=====
package Process_Table

```

```

is new Generic Queue Manager (
  Element_Type => Process_Table_Entry,
  '<=>' => '<=>');

```

```

-- the above package encapsulates the process table; new entries in the
-- process table are added to this list as processes are declared (via
-- declare_process) or as the Kernel notes a process was created on another
-- node (via a Kernel-to-Kernel Initialization protocol message)

```

```

pragma (page);

```

```

function Create_Process_Information_Record return Process_Identifier;

```

```

--/ SUBPROGRAM NAME:

```

```

--/ create_process_information_record

```

```

--/ DESCRIPTION:

```

```

--/ creates a new process information record, returns a pointer to it.
--/ (this may be a direct call on the Ada allocator or may shield a call
--/ to a Kernel-provided storage manager)

```

```

--/ --REQ: 5.1.4

```

```

--/ --PRIM: N/A

```

```

--/ --TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ function return value

```

```

--/ pointer to a new process information record

```

```

--/ PRECONDITIONS:

```

```

--/ none

```

```

--/ ACTIONS:

```

```

--/ create a new process_information_record and return a pointer to it

```

```

--/ POSTCONDITIONS:

```

```

--/ the storage is allocated and returned to the caller

```

```

--/ ERROR PROCESSING:

```

```

--/ when the allocator fails due to there not being sufficient storage

```

```

--/ available:

```

```

--/ the exception insufficient_space is raised to be raised in the

```

```

--/ calling process

```

```

--/ POSTCONDITIONS:

```

```

--/ the requested storage must be assumed to not have been

```

```

--/ allocated

```

```

--/ =====
pragma (page);

```

```

procedure Destroy Process Information Record (
  Process Id : in out Process Identifier);
--=====
--/ SUBPROGRAM NAME
--/ destroy process_information_record
--/
--/ DESCRIPTION:
--/ destroys a specified process information record, sets the pointer to
--/ it to null process
--/
--/ --REQ: 5.1.4
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ process ID
--/ pointer to a process_information_record to destroy
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ destroy a process information record and set the pointer to it to
--/ null process
--/
--/ POSTCONDITIONS:
--/ the storage is returned to the storage manager
--/
--/ ERROR PROCESSING:
--/ none
--/
--/ =====
pragma (page);

```

```

procedure Initialize Process Table (
  Initial Allocation : in t4w Natural);
--=====
--/ SUBPROGRAM NAME:
--/ Initialize process_table
--/
--/ DESCRIPTION:
--/ sets the initial number of entries in the process table to the
--/ specified value
--/
--/ --REQ: 5.1.4
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Initial allocation
--/ number of entries to initially allocate
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ allocates the specified number of process table entry records
--/ creates a process table entry for the null process
--/
--/ POSTCONDITIONS:
--/ the storage is allocated
--/ the null process is initialized and entered into the process table
--/
--/ ERROR PROCESSING:
--/ when the null process process table entry cannot be created:
--/ the Kernel raises insufficient space
--/ POSTCONDITIONS:
--/ the storage is not allocated; process table initialization
--/ fails (this should never happen)
--/
--/ =====
pragma (page);

```

```

function Locate_Process (Name: in Hw_String)
return Process_Identifier;
--/
--/
--/ SUBPROGRAM NAME:
--/ locate_process
--/
--/ DESCRIPTION:
--/ Find the process table entry of a process by searching for its
--/ logical name in the process table.
--/
--/ --REQ: 5.1.4
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ name
--/ logical name of the process to find
--/
--/ default value:
--/ none
--/
--/ return
--/ process table index of the process
--/
--/ values:
--/ if found: a legal process index
--/ if not found: null
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ Locate the process in the process table
--/
--/ POSTCONDITIONS:
--/ None
--/
--/ ERROR PROCESSING:
--/ None
--/
--/
pragma (page);

```

```

function Size_Of_Process_Table return Hw_Natural;
pragma Inline (Size_Of_Process_Table);
--/
--/
--/ SUBPROGRAM NAME:
--/ size_of_process_table
--/
--/ DESCRIPTION:
--/ Returns the number of process entries in the process table
--/ (at the time of invocation)
--/
--/ --REQ: 5.1.4; 14.1.15
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ return
--/ the size of the process table
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ none
--/
--/ ACTIONS:
--/ return the size of the process table
--/
--/ POSTCONDITIONS:
--/ None
--/
--/ ERROR PROCESSING:
--/ None
--/
--/
end Generic_Process_Table;
pragma (page);

```

```

.....
./ MODULE NAME:
./ process_table
./
./ MODULE TYPE:
./ package specification
./
./ MODULE PURPOSE:
./ an example of (and a template for) an instantiation of the generic
./ package Generic_process_table
./
./ MODULE DESCRIPTION:
./ package instantiating the definition of the process table data structure
./ where
./ maximum_number_of_processes_value is defined as 100
./
./ MODULE CONTENTS:
./ process_table (package specification)
./ acknowledged_message_information (type)
./ adjacent_process_table_entry (type)
./ communication_attributes_information (type)
./ create_process_information_record (function)
./ destroy_process_information_record (procedure)
./ event_identifier (subtype)
./ event_information (type)
./ initialize_process_table (procedure)
./ locate_process (function)
./ maximum_number_of_processes (constant)
./ null_event (constant)
./ null_process (logical constant)
./ null_semaphore (constant)
./ pending_activity_attributes_information (type)
./ pending_activity_type (type)
./ process_attributes_information (type)
./ process_identifier (type)
./ process_information_record (type)
./ process_initialization_status_type (type)
./ process_table (generic package instantiation)
./ process_table_entry (type)
./ process_type (type)
./ schedule_attributes_information (type)
./ semaphore (type)
./ semaphore_head (type)
./ semaphore_head_ptr (type)
./ semaphore_attributes_information (type)
./ time_event_queue (generic package instantiation)
./ tool_interface_attributes_information (type)
./ "<" (overloaded (2) function)
./
./ REFERENCES:
./ DESIGN DOCUMENTS:
./ none
./
./ USER'S MANUAL:

```

```

./ Kernel User's Manual
./
./ TESTING AND VALIDATION:
./ none
./
./ NOTES:
./ none
./
./ MODIFICATION HISTORY:
./ 18aug88 bamberg created (complies clean)
./ 25aug88 bamberg final doc. check for 01sep89 freeze
./
./ VERSION: 3.0
./
./ DISCLAIMER:
./ The following copyright must be included in this software and
./ all software utilizing this software.
./
./ DISTRIBUTION:
./ Approved for public release; distribution is unlimited.
./
./ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
./ PA. The Software Engineering Institute (SEI) is a federally
./ funded research and development center established and operated
./ by Carnegie Mellon University (CMU). Sponsored by the U.S.
./ Department of Defense under contract F19628-85-C-0003, the SEI
./ is supported by the services and defense agencies, with the U.S.
./ Air Force as the executive contracting agent.
./
./ Permission to use, copy, modify, or distribute this software and
./ its documentation for any purpose and without fee is hereby
./ granted, provided that the above copyright notice appear in all
./ copies and that both that copyright notice and this permission
./ notice appear in supporting documentation. Further, the names
./ of Software Engineering Institute or Carnegie Mellon University may
./ not be used in advertising or publicity pertaining to
./ distribution of the software without specific, written prior
./ permission. CMU makes no claims or representations about the
./ suitability of this software for any purpose. This software is
./ provided "as is" and no warranty, express or implied, is made
./ by the SEI or CMU, as to the accuracy and functioning of the
./ program and related program material, nor shall the fact of
./ distribution constitute any such warranty. No responsibility is
./ assumed by the SEI or CMU in connection herewith.
./ .....

```

```

with Generic Process Table;
pragma Elaborate (Generic_Process_Table);
-- to get: see MODULE CONTENTS

```

```

../package short name: PTB
.....
package Process_Table

le new Generic_Process_Table (
Maximum Number Of Processes Value => 25);
pragma page);

.....
../MODULE NAME:
../ generic_tool_interface
../
../MODULE TYPE:
../ generic_package_specification
../
../MODULE PURPOSE:
../ provide non-intrusive read-only access to Kernel Internal data
../ structures
../
../MODULE DESCRIPTION:
../ collection of subprograms and data structure templates that provide
../ a view of the Kernel Internal data structures
../
../MODULE CONTENTS:
../ tool_interface (generic package specification)
../ begin_collection (procedure)
../ cease_collection (procedure)
../ first_n_primitive_name_chars (generic formal parameter)
../ full_process_id (type)
../ interrupt_table (type)
../ kernel_primitive_name_type (type)
../ message_attributes_entry (type)
../ message_attributes_entry_pointer (type)
../ message_contents_type (type)
../ message_length (array)
../ process_attributes_entry (type)
../ process_attributes_entry_pointer (type)
../ process_table (type)
../ read_interrupt_table (procedure)
../ size_of_process_table (function)
../ to_message_attributes (function)
../ to_process_attributes (function)
../
../REFERENCES:
../ DESIGN DOCUMENTS:
../ Kernel Facilities Definition
../ --REQ: 14.1.6, 14.1.9, 14.1.10, 14.1.12
../ --REQ: 14.1.13, 14.1.15, 14.1.16
../ --PRIM: N/A
../ --TIME: N/A
../
../ USER'S MANUAL:
../ Kernel User's Manual
../
../ TESTING AND VALIDATION:
../ Unit testing
../
../NOTES:
../ none
../

```



```

--/ MODIFICATION HISTORY:
--/ 18apr88 bamborg created (compiles clean)
--/ 12may89 nys Implement the tool interface
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/
--/ Approved for public release, distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....

```

pragma (page);

```

with Communication_Globals;
-- to get: message_tag_type;

with Hardware_Interface; use Hardware_Interface;
-- to get: hw_address
-- hw_integer
-- hw_positive
-- hw_string

with Interrupt_Globals;
-- to get: interrupt_condition
-- interrupt_state

with Kernel_Exceptions;
-- to get: kernel_exceptions

with Network_Globals;
-- to get: bus_address

with Process_Table;
-- to get: process_identifier

with Schedule_Types;
-- to get: preemption
-- priority
-- process_state

with Time_Globals;
-- to get: elapsed_time

with Tool_Interface_Globals;
-- to get: collection_type

with Unchecked_Conversion;

generic

-- the number of characters that are saved in the tables for the Kernel
-- primitive_name

First_N_Primitive_Name_Chars: In Hw_Positive := 32;

Pragma(Page);
-- .....
-- package short name: GTI
-- .....

package Generic_Tool_Interface

```

```

is
package Cg
renames Communication_Globals;
package Ig
renames Interrupt_Globals;
package Ke
renames Kernel_Exceptions;
package Ng
renames Network_Globals;
package Pb
renames Process_Table;
package St
renames Schedule_Types;
package Tg
renames Time_Globals;
package Tig
renames Tool_Interface_Globals;

.. type of a string sufficiently long to hold the name of a Kernel primitive

type Kernel_Primitive_Name_Type is new Hw_String (
  1 .. Positive (First N_Primitive_Name_Chars) );
.. [***NOTE: UGH!!! I don't like this ... must I convert
.. w/positive here????? **]

pragma (page);

.. the process attributes are available on a per-process basis
.. each entry in the process attributes data structure includes the
.. following information:
..   id      the Kernel-generated process_id
..   state   an indication of whether the process (id) is: running, suspended,
..           blocked, or dead
..   time_state_change the elapsed time when the process (id) entered the above state
..   current_priority the priority level at which the process (id) is currently
..                   registered
..   current_preemption indication of whether or not the process (id) is preemptible
..   alarm         indication of whether or not an alarm is set for the process (id)
..   primitive_identity name of the Kernel primitive invocation by the process (id)
..   primitive_return_status indication of the status of the above Kernel primitive call by
..                   the process (id)
.. REQ: 14.1.6, 14.1.9

type Process_Attributes_Entry is record
  Id: Ptb.Process_Identifier;
  State: St.Process_State;
  Time_State_Change: Tg.Epoch_Time;
  Current_Priority: St.Priority;
  Current_Preemption: St.Preemption;
  Alarm_Pending: Boolean;
  Primitive_Identity: Kernel_Primitive_Name_Type;
  Primitive_Return_Status: Ke.Kernel_Exceptions;
end record;

.. This type and unchecked conversion function allow the user of the
.. tool interface to cast the message buffer address used by the tool
.. interface process into a pointer to a record containing the data.

type Process_Attributes_Entry_Pointer is access Process_Attributes_Entry;
function To_Process_Attributes is new Unchecked_Conversion
  (Source => Hw_Address,
   Target => Process_Attributes_Entry_Pointer);

pragma (page);

```

```

Target => Message_Attributes_Entry_Pointer);
.. the message contents is an array of bytes
.. REQ: 14.1.13,
..
type Message_Contents_Type is array (hw Natural range <=>) of hw Byte;
pragma (page);

```

```

.. this type represents the full process id as it would appear on the
.. processor on which the process is siled
..
.. there are two parts to this information:
..   siled_processor_id
..   the bus address of the processor on which the process is siled
..   process_id
.. the process id of the process on the processor on which it is
..   siled (i.e., its local process id on its owning processor)
.. REQ: 14.1.10

```

```

type Full_Process_Id is record
  Process_Id: Ptb_Process_Identifier;
  Siled_Processor_Id: Ng_Bus_Address;
end record;

```

```

.. each entry in the message attributes data structure includes the
.. following information
..   sender_process_id
..   the full id of the sending process
..   receiver_process_id
..   the full id of the receiving process
..   time_kernel_got_message
..   for a sent message, this is time when Kernel SEND gets control
..   for a received message, this is time when Kernel receives that
..   message. NOT when the Kernel RECEIVE is called
..   message_length
..   the number of bytes in the message text itself
..   message_tag
..   the application-specified tag indicating the kind of message this
..   is
.. REQ: 14.1.10; 14.1.12

```

```

type Message_Attributes_Entry is record
  Sender_Process_Id: Full_Process_Id;
  Receiver_Process_Id: Full_Process_Id;
  Message_Length: Cg_Message_Length_Type;
  Message_Tag: Cg_Message_Tag_Type;
  Time_Kernel_Got_Message: Ig_Epoch_Time;
end record;

```

```

.. This type and unchecked conversion function allow the user of the
.. tool interface to cast the message buffer address used by the tool
.. interface process into a pointer to a record containing the data

```

```

type Message_Attributes_Entry_Pointer is access Message_Attributes_Entry;
function To_Message_Attributes is new Unchecked_Conversion
(Source => hw_Address);

```

```

-- the process table is available on a per-processor basis
--
-- REQ: 14.1.15
--
type Process_Table is array (hw_Natural range <>) of
  Ptb.Process_Information_Record;

-- the interrupt table records information about interrupts and is
-- available on a per-processor basis
--
-- REQ: 14.1.16
--
type Interrupt_Table is new lg Interrupt_Table_Type;

Message_Length : array (Tlg_Collection_Type) of Cg_Message_Length_Type :=
  (Tlg_Process_Attributes =>
    Cg_Message_Length_Type(Process_Attributes_Entry.Size/Bits_Per_Byte),
    Tlg_Message_Attributes =>
    Cg_Message_Length_Type(Message_Attributes_Entry.Size/Bits_Per_Byte),
    Tlg_Message_Contents =>
    Cg_Message_Length_Type(Cg_Maximum_Message_Length));
pragma (page);

```

```

procedure Begin_Collection (Process_Id: In Ptb.Process_Identifier;
  Tool_Message_Tag: In Cg_Message_Tag_Type;
  Information_To_Collect: In Tlg_Collection_Type);
--
-- SUBPROGRAM NAME:
-- begin_collection
--
-- DESCRIPTION:
-- enables the logging of information for the specified process on the
-- specified type of collection
--
-- REQ: 14.1.1; 14.1.2; 14.1.3; 14.1.4; 14.1.6; 14.1.7; 14.1.9;
-- REQ: 14.1.10; 14.1.12; 14.1.13; 14.1.18
-- PRIM: 23.1.3
--
-- TIME: 14.2.1; 14.2.4
--
PARAMETERS:
-- process_id
-- the id of the process on which collection is to be performed
--
-- default_value:
-- none
--
-- tool_message_tag
-- the message tag to accompany the tool interface message containing
-- the indicated attributes for the indicated process
--
-- values_include:
-- any legal message tag
--
-- default_value:
-- none
--
-- information_to_collect
-- the kind of information to be collected for the specified process
--
-- values_include:
-- process_attributes
-- information about the process identity and state
-- message_attributes
-- the message "envelope" (e.g., sender, receiver, message tag,
-- message length)
-- message_contents
-- the actual message text
--
-- default_value:
-- none
--
PRECONDITIONS:
-- NONE
--
ACTIONS:
-- enables collection of the specified process attributes for the

```

```

--/ specified process
--/
--/ POSTCONDITIONS:
--/ enabled collection information
--/
--/ ERROR PROCESSING:
--/ none: all erroneous collection requests are simply ignored
--/
pragma (page);

```

```

procedure Cease_Collection (Process_Id: In Pib.Process_Identifier;
Information_No: To Collect: In Tlg.Collection_Type);
--/
--/ SUBPROGRAM NAME:
--/ cease collection
--/
--/ DESCRIPTION:
--/ disables the logging of information for the specified process on the
--/ specified type of collection
--/
--/ REQ: 14.1.8; 14.1.11; 14.1.14; 14.1.18
--/
--/ PRIM: 23.1.4
--/
--/ TIME: 14.2.2; 14.2.4
--/
--/ PARAMETERS:
--/ process_id
--/ the id of the process on which collection is to be disabled
--/
--/ default value:
--/ none
--/
--/ Information to collect
--/ the kind of information for which collecting is to be disabled
--/ for the specified process
--/
--/ values include:
--/ process_attributes
--/ information about the process identity and state
--/ message_attributes
--/ the message "envelope" (e.g., sender, receiver, message tag,
--/ message length)
--/ message_contents
--/ the actual message text
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ disables collection of the specified process attributes for the
--/ specified process
--/
--/ POSTCONDITIONS:
--/ disabled collection information
--/
--/ ERROR PROCESSING:
--/ none: all erroneous collection requests are simply ignored
--/

```

pragma (page);

```

function Size_Of_Process_Table return HW_Natural;
--/
--/ SUBPROGRAM NAME:
--/ size_of_process_table
--/
--/ DESCRIPTION:
--/ extracts the size of the Kernel's process table
--/
--/ --REQ: 14.1.15
--/ --PRIM: 23.1.7
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ return
--/ the size of the process table
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ Return the size of the process table
--/
--/ POSTCONDITIONS:
--/ none
--/
--/ ERROR PROCESSING:
--/ none
--/
--/
pragma (page);

```

```

procedure Read_Process_Table (The_Process_Table: out Process_Table;
Last_Entry: out HW_Natural);
--/
--/ SUBPROGRAM NAME:
--/ read_process_table
--/
--/ DESCRIPTION:
--/ extracts information from the kernel's process table
--/
--/ --REQ: 14.1.15
--/ --PRIM: 23.1.5
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ the process table
--/ the table to hold a snapshot of the process table
--/
--/ default value:
--/ none
--/
--/ last entry
--/ the last location of meaningful data in the process_table
--/
--/ default value:
--/ none
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ reads the process table information into a user-provided buffer
--/
--/ POSTCONDITIONS:
--/ the user-provided buffer contains as much of the process table as
--/ will fit into the user-provided buffer
--/ no change in kernel data structures
--/
--/ ERROR PROCESSING:
--/ none
--/
--/
pragma (page);

```

```

procedure Read Interrupt Table (The Interrupt Table; out Interrupt Table);

```

```

--SUBPROGRAM NAME:
--read Interrupt table

```

```

--DESCRIPTION:
--extracts information from the Kernel's Interrupt table

```

```

--REQ: 14.1.16

```

```

--PRIM: 23.1.6

```

```

--TIME: N/A

```

```

--PARAMETERS:

```

```

--the Interrupt table
--the location where the Kernel is to dump the Interrupt table
--information

```

```

--default value:

```

```

--none

```

```

--PRECONDITIONS:

```

```

--NONE

```

```

--ACTIONS

```

```

--reads the Interrupt table information into a user-provided buffer

```

```

--POSTCONDITIONS:

```

```

--the user-provided buffer contains as much of the Interrupt table as
--will fit into the user-provided buffer
--no change in Kernel data structures

```

```

--ERROR PROCESSING:

```

```

--none

```

```

end Generic Tool Interface;
pragma (page);

```

```

--MODULE NAME:
--tool_interface
--MODULE TYPE:
--package specification
--MODULE PURPOSE:
--an example of (and a template for) an instantiation of the generic
--package generic_tool_interface
--MODULE DESCRIPTION:
--collection of subprograms and data structure templates that provide
--a view of the Kernel internal data structures
--MODULE CONTENTS:
--tool_interface (package specification)
--begin_collection (procedure)
--cease_collection (procedure)
--first_n_primitive_name_chars (generic formal parameter)
--full_process_id (type)
--Interrupt_table (type)
--Kernel_primitive_name_type (type)
--message_attributes_entry (type)
--message_attributes_entry_pointer (type)
--message_contents_type (type)
--message_length (array)
--process_attributes_entry (type)
--process_attributes_entry_pointer (type)
--process_table (type)
--read_process_table (procedure)
--read_interrupt_table (procedure)
--size_of_process_table (function)
--to_message_attributes (function)
--to_process_attributes (function)
--REFERENCES:
--DESIGN DOCUMENTS:
--Kernel Facilities Definition
--USER'S MANUAL:
--Kernel User's Manual
--TESTING AND VALIDATION:
--none
--NOTES:
--none
--MODIFICATION HISTORY:
--18Aug88 bamberg created (complies clean)

```

```

--/ VERSION: 3.0
--/ .....
--/ DISCLAIMER:
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/ DISTRIBUTION:
--/ Approved for public release; distribution is unlimited.
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....
--/ with Generic Tool Interface;
--/ pragma Elaborate (Generic Tool Interface);
--/ ... to get: see MODULE CONTENTS
--/ .....
--/ package short name: Ti
--/ .....
--/ package Tool Interface
--/ is new Generic Tool Interface
--/ (
--/ First N Primitive Name Chars > 32
--/ ),
--/ pragma (page);
--/ .....
--/ MODULE NAME:
--/ tool interface_output
--/ .....
--/ MODULE TYPE:
--/ package specification
--/ .....
--/ MODULE PURPOSE:
--/ conditional compilation flag indicating whether or not the code for
--/ output of the tool interface information should be included in the
--/ compilation output or not
--/ .....
--/ MODULE DESCRIPTION:
--/ the conditional compilation flag
--/ .....
--/ MODULE CONTENTS:
--/ tool interface_output (package specification)
--/ tool interface_output_enabled (constant)
--/ .....
--/ REFERENCES:
--/ DESIGN DOCUMENTS:
--/ Kernel Facilities Definition
--/ --REQ: 5.1.7
--/ --PRIM: N/A
--/ --TIME: N/A
--/ .....
--/ USER'S MANUAL:
--/ Kernel User's Manual
--/ .....
--/ TESTING AND VALIDATION:
--/ Integration testing with generic tool interface
--/ .....
--/ NOTES:
--/ this is included (via a with clause) in the appropriate bodies of the
--/ Kernel to enable/disable compilation of Kernel code (note that the
--/ exploitation of conditional compilation based on constant values is a
--/ compiler dependency); the Kernel user may change the setting of this
--/ flag, but must pay the price of recompilation of all Kernel bodies (in
--/ the worst case); no Kernel specifications are affected by this
--/ .....
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (complies clean)
--/ .....
--/ VERSION: 3.0
--/ .....
--/ DISCLAIMER:
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/ DISTRIBUTION:

```

357

December 1989

December 1989

/ Approved for public release, distribution is unlimited.

Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh, PA. The Software Engineering Institute (SEI) is a federally funded research and development center established and operated by Carnegie Mellon University (CMU). Sponsored by the U.S. Department of Defense under contract F19628-85-C-0003, the SEI is supported by the services and defense agencies, with the U.S. Air Force as the executive contracting agent.

Permission to use, copy, modify, or distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Further, the names of Software Engineering Institute or Carnegie Mellon University may not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. CMU makes no claims or representations about the suitability of this software for any purpose. This software is provided "as is" and with no warranty, express or implied, by the SEI or CMU, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty. No responsibility is assumed by the SEI or CMU in connection herewith.

```
...../ ..... package short name: TKO .....
```

Package Tool Interface Output

1

```
.. this flag is used to enable/disable conditional compilation of all tool
.. interface code
```

```

Tool Interface Output Enabled : constant Boolean := True;

end Tool Interface Output;

pragma page;

```

[illegible]

```
prgno (page):
```

```
...../ ..... package short name: TKO .....
```

Package Tool Interface Output

1

```
.. this flag is used to enable/disable conditional compilation of all tool
.. interface code
```

```

Tool Interface Output Enabled : constant Boolean := True;

end Tool Interface Output;

pragma page;

```

```

.....
MODULE NAME
  hardware interface

MODULE TYPE
  package specification

MODULE PURPOSE
  The Kernel package hardware interface
  provides an interface to compiler specific primitive types.
  Within the Kernel itself, there are no references to the
  predefined types in Ada package Standard;
  all references to primitive types use names declared in
  package hardware interface.
  By doing this, certain implementation dependent details are abstracted
  away from both the Kernel and the application in a uniform manner.
  It is recommended that applications avoid using package Standard entirely
  and use package hardware interface for ready compatibility with
  Kernel primitives.

MODULE DESCRIPTION:
  these types are defined in case the compiler's definition and/or
  implementation of them is not sufficient for the Kernel and/or the
  application; this allows Kernel/application developers to provide
  their own version of types and operators without needing to modify
  Kernel or application code directly

MODULE CONTENTS:
  hardware interface (package specification)
  bits per byte (constant)
  byte (constant)
  hw address (type)
  hw bit8 (type)
  hw bit8_ptr (type)
  hw byte (type)
  hw byte_ptr (type)
  hw duration (type)
  hw integer (type)
  hw long integer (type)
  hw long natural (type)
  hw long positive (type)
  hw long unsigned (type)
  hw natural (type)
  hw positive (type)
  hw short integer (type) (CURRENTLY COMMENTED OUT)
  hw string (type)
  longword (constant)
  null hw address (constant)
  to hw address (function)
  to hw bit8 (function)
  to hw bit8_ptr (function)
  to hw byte_ptr (function)
  word (constant)

```

```

REFERENCES:
  DESIGN DOCUMENTS:
  Kernel Facilities Definition
  REQ: 5.1.4
  PRIM: N/A
  TIME: N/A

USER'S MANUAL:
  Kernel User's Manual

TESTING AND VALIDATION:
  none

NOTES:
  none

MODIFICATION HISTORY:
  18aug88 bamberg created (complies clean)
  30aug88 bamberg correct typo on bound of hw_integer
  31aug88 dvk changed "byte" to "bits per byte"
  added constants byte, word, and longword
  14sep88 dvk added hw_long natural
  17oct88 ltrhw added hw_byte, hw_bit8, hw_byte_ptr,
  hw_bit8_ptr and conversions
  1Nov88 firth added to hw_address and null hw_address
  12Dec88 firth revised documentation, reorganised
  11Jan89 firth final doc. check for beta release
  19Jul89 dvk added hw_long unsigned
  25aug89 bamberg final doc. check for 01sep89 freeze

VERSION: 3.0

DISCLAIMER:
  The following copyright must be included in this software and
  all software utilizing this software.

DISTRIBUTION:
  Approved for public release, distribution is unlimited.

Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
PA. The Software Engineering Institute (SEI) is a federally
funded research and development center established and operated
by Carnegie Mellon University (CMU). Sponsored by the U.S.
Department of Defense under contract F19628-85-C-0003, the SEI
is supported by the services and defense agencies, with the U.S.
Air Force as the executive contracting agent.

Permission to use, copy, modify, or distribute this software and
its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all
copies and that both that copyright notice and this permission

```

```

--/ notice appear in supporting documentation. Further, the names
--/ Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/ .....

```

pragma (page);

```

with System;
-- to get: address
--           unsigned_longword

with Unchecked_Conversion;

--/ .....
--/ package short name: HI
--/ .....
package Hardware_Interface
is
--REQ: 5.1.4

--/ The following definitions might need modification after any change to
--/ either compiler or hardware. The purpose of the definitions is to
--/ provide a coordinated and clean view of the underlying basic machine
--/ data types and addressing structure.

-- define the bit interface to the underlying system
--

Bits_Per_Byte : constant := 8;

-- define constants to be used as offsets in rep specs; these values
-- indicate how many bytes in the named storage unit (i.e., byte, word,
-- longword)

-- a byte contains 1 byte
-- a word contains 2 bytes
-- a longword contains 4 bytes

Byte : constant := 1;
Word : constant := 2;
Longword : constant := 4;

-- The type hw_duration must capture exactly the hardware
-- representation of the current Ada type DURATION

-- to be defined for hw_duration:
-- range
-- size
-- small (i.e., granularity of measure)

-- In the event of any mismatch, the compiler should reject the
-- representation clauses

type Hw_Duration is new Duration range -86_400.0 .. +86_400.0;
for Hw_Duration'Small use 2.0**(14);
for Hw_Duration'Size use 4 * Bits_Per_Byte;
pragma (page);

```

```

.. The types hw_byte and hw_bits8 are to be used to define essentially
.. untyped data formats, such as those found in device interfaces.
..
.. hw_byte is intended to be an unsigned 8 bit byte
.. hw_bits8 is intended to be a byte considered as a sequence of bits
..
.. Note especially that the bit numbering in the type hw_bits8 should
.. be whatever is most convenient for subsequent use of the type. If
.. the bulk of the hardware and device documentation numbers the bits
.. a certain way, that way should be used in the type declaration.

```

```

type Hw_Byte is range 0..255;
.. for hw_byte's size use 1*bits_per_byte;

```

```

type Hw_Bits8 is record

```

```

  Bit0 : Boolean;
  Bit1 : Boolean;
  Bit2 : Boolean;
  Bit3 : Boolean;
  Bit4 : Boolean;
  Bit5 : Boolean;
  Bit6 : Boolean;
  Bit7 : Boolean;
end record;

```

```

for Hw_Bits8 use record

```

```

  Bit0 at 0 range 0..0;
  Bit1 at 0 range 1..1;
  Bit2 at 0 range 2..2;
  Bit3 at 0 range 3..3;
  Bit4 at 0 range 4..4;
  Bit5 at 0 range 5..5;
  Bit6 at 0 range 6..6;
  Bit7 at 0 range 7..7;
end record;

```

```

pragma (page);

```

```

..
..-----
.. SUBPROGRAM NAME:
.. to_hw_bits8
..
.. DESCRIPTION:
.. conversion routine from hw_byte to hw_bits8
..
.. REQ: 5.1.4
.. PRIM: N/A
.. TIME: N/A
..
.. PARAMETERS:
.. Source
.. an 8-bit value of type hw_byte
.. (a small integer in the range 0..255)
..
.. default value:
.. none
..
.. function return value
.. the same bit pattern, considered as a record of eight
.. single-bit components
..
.. PRECONDITIONS:
.. NONE
..
.. ACTIONS:
.. return the same physical bit pattern as the parameter
..
.. POSTCONDITIONS:
.. NONE
..
.. ERROR PROCESSING:
.. NONE
..
.. EXAMPLES:
.. This conversion is to be used when it is necessary to inspect
.. an individual bit of a device register or other hardware value.
..
.. Consider for instance a device status byte of which Bit2 is the
.. 'ready' bit, asserted whenever the device is ready. The test
.. in Ada then looks like this:
..
.. if to_hw_bits8(status_byte).bit2 then -- device is ready
..
..-----

```

```

function To_Hw_Bits8 is new Unchecked_Conversion(hw_Byte, hw_Bits8);
pragma (page);

```

-- The following types and conversions are intended to allow the
 -- code to set individual bits in device registers. This is
 -- achieved by
 --
 -- (a) declaring the register to be a hw_byte
 -- (b) compiling its Address
 -- (c) casting that address into an access value designating an
 -- object of type hw_bits8
 -- (d) accessing the indivisible bits of that object
 --
 -- these types are pointers to hw_byte or hw_bits8 objects
 -- they are intended to allow unchecked conversion of values

type hw_byte_ptr is access hw_byte;
 type hw_bits8_ptr is access hw_bits8;

```
--
-- SUBPROGRAM NAME:
-- to hw_byte_ptr
--
-- DESCRIPTION:
-- conversion routine from Ada address type to access type
--
-- REQ: 5.1.4
-- PRIM: N/A
-- TIME: N/A
--
-- PARAMETERS:
-- Source
-- the address of a declared object
-- WHICH MUST BE A HW_BYTE
--
-- default value:
-- none
--
-- function return value
-- the same bit pattern, considered as an access value designating
-- a hw_byte
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- return the same physical bit pattern as the parameter
--
-- POSTCONDITIONS:
-- NONE
--
-- ERROR PROCESSING:
-- NONE
--
-- EXAMPLES:
-- This conversion is to be used when it is necessary to change
-- an individual bit of a device register or other hardware object.
-- Consider for instance a device status byte of which Bit4 is the
-- 'enable' bit, to be set to enable the device. The code
-- in Ada then looks like this.
--
-- -- to enable the device, we must set Bit 4 of the status byte
-- to hw_bits8_ptr(status_byte'address).bit4 := True;
```

```
function To_hw_byte_ptr is new Unchecked_Conversion(System.Address, hw_byte_ptr)
--
-- SUBPROGRAM NAME:
-- to_hw_bits8_ptr
--
-- DESCRIPTION:
-- conversion routine from Ada address type to access type
--
-- REQ: 5.1.4
-- PRIM: N/A
-- TIME: N/A
--
-- PARAMETERS:
-- Source
-- the address of a declared object
-- WHICH MUST BE A HW_BYTE OR A HW_BITS8
--
-- default value:
-- none
--
-- function return value
-- the same bit pattern, considered as an access value designating
-- a hw_bits8
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- return the same physical bit pattern as the parameter
--
-- POSTCONDITIONS:
-- NONE
--
-- ERROR PROCESSING:
-- NONE
--
-- EXAMPLES:
-- This conversion is to be used when it is necessary to change
-- an individual bit of a device register or other hardware object.
-- Consider for instance a device status byte of which Bit4 is the
-- 'enable' bit, to be set to enable the device. The code
-- in Ada then looks like this.
--
-- -- to enable the device, we must set Bit 4 of the status byte
-- to hw_bits8_ptr(status_byte'address).bit4 := True;
```

```
function To_hw_bits8_ptr is new Unchecked_Conversion(System.Address, hw_bits8_ptr)
```

```

--/ -----
--/ SUBPROGRAM NAME:
--/   to_hw_bits8_ptr
--/
--/ DESCRIPTION:
--/   conversion routine from Ada address type to access type
--/
--/ --REQ: 5.1.4
--/ --PRIM: N/A
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/   Source
--/     the address of a declared object
--/     WHICH MUST BE A HW_BYTE OR A HW_BITS8
--/   default value:
--/     none
--/
--/ function return value
--/   the same bit pattern, considered as an access value designating
--/   a hw_bits8
--/
--/ PRECONDITIONS:
--/   NONE
--/
--/ ACTIONS:
--/   return the same physical bit pattern as the parameter
--/
--/ POSTCONDITIONS:
--/   NONE
--/
--/ ERROR PROCESSING:
--/   NONE
--/
--/ EXAMPLES:
--/   see above
--/ -----

```

```

function To_Hw_Bits8_Ptr is new Unchecked_Conversion(Hw_Byte_Ptr,Hw_Bits8_Ptr);

```

pragma (page);

```

-- The Integer types are intended to capture the natural efficient
-- Integer types of the target machine. Currently, these are
--
--   8-bit signed : hw_short_Integer
--   16-bit signed : hw_Integer
--   32-bit signed : hw_long_Integer
--
-- These types are defined in terms of explicit ranges, and a size
-- clause is provided as a check that the compiler has indeed mapped
-- them onto the correct hardware type.
--
-- The Ada types define, in package STANDARD should not be used, since
-- they are dependent on the compiler as well as the target
--
-- hw_Integer is intended to be a 16-bit Integer
--
-- to be defined for hw_Integer:
--
--   range
--   size
--
-- type hw_Integer is range -32_768 .. 32_767;
-- for hw_Integer'Size use 2 * Bits_Per_Byte;
--
-- hw_short_Integer is intended to be an 8-bit Integer
--
-- to be defined for hw_short_Integer:
--
--   range
--   size
--
-- type hw_short_Integer is range -128 .. 127;
-- for hw_short_Integer'Size use 1 * Bits_Per_Byte;
--
-- hw_long_Integer is intended to be a 32-bit Integer
--
-- to be defined for hw_long_Integer:
--
--   range
--   size
--
-- type hw_long_Integer is range -2_147_483_648 .. 2_147_483_647;
-- for hw_long_Integer'Size use 4 * Bits_Per_Byte;
-- pragma (page);

```

```

.. For additional security, the appropriate subsets of the machine
.. integer types are defined as full types, not as subtypes. The
.. two subsets are
..
.. Natural: the non-negative half of the range (0..Last)
.. Positive: the strictly positive subset (1..Last)
..
.. Again, a size clause is added as a check.
..
.. hw_long_natural is intended to be a 32-bit integer with natural range
..
.. to be defined for hw_long_natural:
..
.. range
.. size
..
type Hw_Long_Natural is range 0 .. 2 147 483 647;
for Hw_Long_Natural'Size use 4 * Bits_Per_Byte;

.. hw_long_positive is intended to be a 32-bit integer with positive range
..
.. to be defined for hw_long_positive:
..
.. range
.. size
..
type Hw_Long_Positive is range 1 .. 2 147 483 647;
for Hw_Long_Positive'Size use 4 * Bits_Per_Byte;

.. hw_long_unsigned is intended to be a full unsigned 32-bit integer
..
.. to be defined for hw_long_unsigned:
..
.. range
.. size
..
type Hw_Long_Unsigned is new System.Unsigned Longword;
for Hw_Long_Unsigned'Size use 4 * Bits_Per_Byte;

.. the range of hw_natural is to be a subset of hw_integer
.. (i.e., only 16-bits)
..
.. to be defined for hw_natural:
..
.. range
.. size
..
type Hw_Natural is range 0 .. 32 767;
for Hw_Natural'Size use 2 * Bits_Per_Byte;

.. the range of hw_positive is to be a subset of hw_integer

```

```

.. (i.e., only 16-bits)
..
.. to be defined for hw_positive:
..
.. range
.. size
..
type Hw_Positive is range 1 .. 32 767;
for Hw_Positive'Size use 2 * Bits_Per_Byte;
pragma page;

```

```

.. hw_address is included just in the off chance that the compiler's
.. implementation of system_address is not acceptable to the Kernel and/or
.. the application, this allows Kernel/application developers to provide
.. their own version of address operators without needing to modify Kernel
.. or application code
..
.. There is also a function to convert an integer of the appropriate
.. size to an address, and a null value that cannot be a legal address
..
.. In the current implementation, it has been shown that the
.. compiler provided address type is satisfactory.

```

```

type hw_address is new System_Address;

```

```

..
.. SUBPROGRAM NAME:
.. to_hw_address
.. DESCRIPTION:
.. conversion routine from an integer type to type hw_address
..
.. --REQ: NONE
..
.. --PRIM: NONE
..
.. --TIME: NONE
..
.. PARAMETERS:
.. Source
.. an integer value OF THE SAME SIZE AS AN ADDRESS
.. the value MUST be a legal address
.. default value:
.. none
..
.. function return value
.. the same bit pattern, considered as an address
..
.. PRECONDITIONS:
.. NONE
..
.. ACTIONS:
.. return the same physical bit pattern as the parameter
..
.. POSTCONDITIONS:
.. NONE
..
.. ERROR PROCESSING:
.. NONE
..
.. EXAMPLES:
.. The most common use of this function is to construct a pointer
.. to a specific part of the machine's address space, for instance

```

```

.. a part that corresponds to a device control register. Suppose
.. that a device is controlled by a set of registers whose addresses
.. start at 16#10001ff0#. A handler on this device can be created by
..
.. device control block address : hw_address
.. := to_hw_address(16#10001ff0);
..

```

```

function To_Hw_Address is
newUnchecked_Conversion(hw_Long_Integer, hw_Address);
Null_Hw_Address : constant hw_Address
:= To_Hw_Address(hw_Long_Integer(0));

```

```

.. hw_string is included just in the off chance that the compiler's
.. implementation of string is not acceptable to the Kernel and/or the
.. application, this allows Kernel/application developers to provide their
.. own version of string operators without needing to modify Kernel or
.. application code
..
.. In this initial implementation, we assume that the compiler provided
.. string type and operations are OK

```

```

type hw_string is new String;
end Hardware_Interface;
pragma page;

```



```

--/ MODULE NAME:
--/   generic_kernel_time
--/
--/ MODULE TYPE:
--/   generic_package_specification
--/
--/ MODULE PURPOSE:
--/   provide the abstraction of Kernel time
--/
--/ MODULE DESCRIPTION:
--/   Kernel_time abstract type and subprograms to manipulate it
--/
--/ MODULE CONTENTS:
--/   generic_kernel_time (generic package specification)
--/   integral_duration (type)
--/   Kernel_time (type)
--/   seconds (overloaded (2) function)
--/   milliseconds (function)
--/   ticks_per_second (constant)
--/   ticks_per_second_value (generic formal parameter)
--/   zero_Kernel_time (constant)
--/   ">*" (unary; function)
--/   ">*" (function)
--/   ">*" (unary; function)
--/   ">*" (function)
--/   ">*" (overloaded (2) function)
--/   ">*" (function)
--/   "<*" (function)
--/   "<*" (function)
--/   ">*" (function)
--/   ">*" (function)
--/
--/ REFERENCES:
--/   Kernel Facilities Definition
--/   --REQ: 12.1.1, 12.1.5, 12.1.6, 12.1.25
--/   --PRM: N/A
--/   --TIME: N/A
--/
--/ DESIGN DOCUMENTS:
--/   Kernel Architecture Manual
--/
--/ USER'S MANUAL:
--/   Kernel User's Manual
--/
--/ TESTING AND VALIDATION:
--/   Unit tests and integration testing with Time_keeper
--/
--/ NOTES:
--/   this package must be instantiated and used in place of the
--/   compiler-provided Package_Calendar
--/
--/   this package provides an abstract data type representing Time

```

```

--/ for use by the kernel. It is at the lowest level of a set of
--/ abstractions that jointly respond to Requirements Section 12.
--/
--/ MODIFICATION HISTORY:
--/ 18aug88 bamberg created (complies clean)
--/ 18oct88 ltrh revised to interface to assembler
--/ 18oct88 ltrh changed type of ticks_per... to long int
--/ 24oct88 ltrh added 'milliseconds' and 'microseconds'
--/ 09jan89 ltrh improved documentation
--/ 11jan89 ltrh final doc update for beta release
--/ 16feb89 ltrh corrected error in documentation
--/ 29jun89 dvk shifted to VMS definitions
--/ 19jul89 dvk removed dependency on package system
--/ fixed interface spec for GKT_seconds1
--/
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/ The following copyright must be included in this software and
--/ all software utilizing this software.
--/
--/ DISTRIBUTION:
--/ Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/ PA. The Software Engineering Institute (SEI) is a federally
--/ funded research and development center established and operated
--/ by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/ Department of Defense under contract F19628-85-C-0003, the SEI
--/ is supported by the services and defense agencies, with the U.S.
--/ Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/ its documentation for any purpose and without fee is hereby
--/ granted, provided that the above copyright notice appear in all
--/ copies and that both that copyright notice and this permission
--/ notice appear in supporting documentation. Further, the names
--/ of Software Engineering Institute or Carnegie Mellon University may
--/ not be used in advertising or publicity pertaining to
--/ distribution of the software without specific, written prior
--/ permission. CMU makes no claims or representations about the
--/ suitability of this software for any purpose. This software is
--/ provided "as is" and no warranty, express or implied, is made
--/ by the SEI or CMU, as to the accuracy and functioning of the
--/ program and related program material, nor shall the fact of
--/ distribution constitute any such warranty. No responsibility is
--/ assumed by the SEI or CMU in connection herewith.
--/
--/ with Hardware Interface; use Hardware Interface;
--/ to get: hw_duration

```

```

--      hw_long_integer
--      hw_positive
pragma (page);

```

```

generic
--
-- the number of TICKs in a second (this value is never used within the
-- Kernel except for the initialization of the constant, nor should it be
-- used without the Kernel)
--
    Ticks_Per_Second_Value : in Hw_Long_Integer;
--/ .....
--/ package short name: GKT
--/ .....
package Generic_Kernel_Time
is
--
-- the constant number of TICKs per second (thus providing a mechanism of
-- converting from Kernel time to application time)
--
    Ticks_Per_Second : Hw_Long_Integer := Ticks_Per_Second_Value;
--
-- type Kernel_time: the time on which elapsed time and epoch time
-- abstractions are built
--
-- this time representation allows applications beginning at time zero to
-- execute for some 15_000 years, so no overflow is monitored
-- .....
-- the user should ensure that adjusting any times does not approach the
-- limit of this time representation; proper Kernel functioning is NOT
-- guaranteed if time overflows
-- .....
-- Kernel_Time is represented as a signed 64-bit binary integer,
-- representing a count of hundred-nanoseconds. Hence, a kernel time
-- of 10_000_000 corresponds to one second.
--
-- For the purposes of Ada definition, kernel_time is a record of
-- two components, being respectively the low-order and high-order
-- 32 bits. The high-order component can be correctly declared to
-- be a signed 32-bit integer, of type hw_long_integer. The low-order
-- component, however, is properly an UNSIGNED 32-bit integer, which
-- we have defined in HW_INTERFACE as hw_long_unsigned for VAX Ada
--
-- Applications should use ONLY the exported constructor functions to
-- create values of type kernel_time or of types derived from it.
type Kernel_Time is record
    Low : Hw_Long_Unsigned := 0;

```

```

High: Hw_Long_Integer := 0;
end record;

for Kernel_Time use record
  Low at 0 range 0..31;
  High at 4 range 0..31;
end record;

```

```

-- the value of zero for the Kernel_time abstraction
--

```

```

Zero_Kernel_Time : constant Kernel_Time := (0,0);

```

```

-- the range of durations represented as integral values
--

```

```

type Integral_Duration is new Hw_Long_Integer;

```

```

pragma (page);

```

```

function Seconds (
  Ada_Duration : Hw_Duration)
return Kernel_Time;

-- The pragma INTERFACE and IMPORT_FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

-- SUBPROGRAM NAME:
-- seconds
-- DESCRIPTION:
-- conversion routine from Ada duration type to Kernel_time
-- REQ: 12.1.23, 12.1.24
-- PRIM: N/A
-- TIME: N/A
-- PARAMETERS:
-- Ada duration
-- the time to convert in Ada duration format (note that this limits
-- legal values to 0 to 86400 seconds, 1 day)
-- default value:
-- none
-- function return value
-- the input duration in Kernel_time format (note that this contains
-- time up to 1 day)
-- PRECONDITIONS:
-- NONE
-- ACTIONS:
-- returns the Ada duration time in Kernel_time format
-- POSTCONDITIONS:
-- NONE
-- ERROR PROCESSING:
-- NONE

```

```

pragma (page);

```

```

function Seconds (
  An Integral Duration : Integral_Duration)
  return Kernel_Time;

pragma Interface(Assembly, Seconds);
pragma Import Function (Internal => Seconds,
  External => "GKT_seconds1",
  Parameter_Types => (Ith_Duration),
  Mechanism => (Reference),
  Result_Type => Kernel_Time);

pragma Import Function (Internal => Seconds,
  External => "GKT_seconds2",
  Parameter_Types => (Integral_Duration),
  Mechanism => (Value),
  Result_Type => Kernel_Time);

--/ =====
--/ SUBPROGRAM NAME:
--/ milliseconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Integral milliseconds
--/ the duration in milliseconds to convert to kernel_time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input Integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/ =====
pragma (page);

```

```

function Microseconds (
  Integral_Microseconds : Integral_Duration)
  return Kernel_Time;

pragma Interface(Assembly, Microseconds);
pragma Import Function (Internal => Microseconds,
  External => "GKT_microseconds",
  Parameter_Types => (Integral_Duration),
  Mechanism => (Value),
  Result_Type => Kernel_Time);

```

```

--/ SUBPROGRAM NAME:
--/ seconds
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel_time
--/ --REQ: 12.1.23, 12.1.24
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ Integral_microseconds
--/ the duration in microseconds to convert to kernel_time
--/ default value:
--/ none
--/ function return value
--/ in a input integral duration in Kernel_time format
--/ PRECONDITIONS:
--/ NONE
--/ ACTIONS:
--/ returns the Integral duration time in Kernel_time format
--/ POSTCONDITIONS:
--/ NONE
--/ ERROR PROCESSING:
--/ NONE

```

pragma (page);

```

function "+" (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Kernel_Time;

-- The pragma INTERFACE and IMPORT_FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

--/ SUBPROGRAM NAME:
--/ "+"
--/ DESCRIPTION:
--/ adds two Kernel_time values returning a Kernel_time value
--/ --REQ: 12.1.23, 12.1.24
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ left
--/ left operand of the addition operator
--/ default value:
--/ none
--/ right
--/ right operand of the addition operator
--/ default value:
--/ none
--/ function return value
--/ result of the addition operator
--/ PRECONDITIONS:
--/ NONE
--/ ACTIONS:
--/ takes the mathematical sum of the two Kernel_time values, returning a
--/ properly formatted Kernel_time result
--/ POSTCONDITIONS:
--/ the sum of left and right is computed and returned
--/ ERROR PROCESSING:
--/ NONE

```

pragma (page);

```

function "*" (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Kernel_Time;

-- The pragma INTERFACE and IMPORT_FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

-- SUBPROGRAM NAME:
-- "*"
-- DESCRIPTION:
-- subtracts the right Kernel_time value from the left Kernel_time
-- value returning a Kernel_time value
-- REQ: 12.1.23, 12.1.24
-- PRIM: N/A
-- TIME: N/A
-- PARAMETERS:
-- left
-- left operand of the subtraction operator (subtrahend)
-- default value:
-- none
-- right
-- right operand of the subtraction operator (minuend)
-- default value:
-- none
-- function return value
-- result of the subtraction operator
-- PRECONDITIONS:
-- NONE
-- ACTIONS:
-- takes the mathematical difference of the two Kernel_time values by
-- subtracting the right operand from the left operand, returning a
-- properly formatted Kernel_time result
-- POSTCONDITIONS:
-- the difference of left and right is computed and returned
-- ERROR PROCESSING:
-- NONE

```

pragma (page);

```

function "*" (
  Left : Kernel_Time;
  Right : Hw_Long_Integer)
return Kernel_Time;

-- The pragma INTERFACE and IMPORT_FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

-- SUBPROGRAM NAME:
-- "*"
-- DESCRIPTION:
-- multiplies a Kernel_time value by an integer value returning a
-- Kernel_time value
-- REQ: 12.1.23, 12.1.24
-- PRIM: N/A
-- TIME: N/A
-- PARAMETERS:
-- left
-- left operand of the multiplication operator
-- default value:
-- none
-- right
-- right operand of the multiplication operator
-- default value:
-- none
-- function return value
-- result of the multiplication operator
-- PRECONDITIONS:
-- NONE
-- ACTIONS:
-- takes the mathematical product of the Kernel_time value and the
-- integer, returning a properly formatted Kernel_time result
-- POSTCONDITIONS:
-- the product of left and right is computed and returned
-- ERROR PROCESSING:
-- NONE

```

pragma (page);

```

--/ the product of left and right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```

```

pragma (page);

```

```

function "" (
  Left : Hw_Long_Integer;
  Right : Kernel_Time)
return Kernel_Time;

pragma Interface(Assembly, "");
pragma Import Function (Internal
  External => "GKI_multiply1";
  Parameter_Type => (Kernel_Time, Hw_Long_Integer),
  Mechanism => (Reference, Value),
  Result_Type => Kernel_Time);

pragma Import Function (Internal
  External => "GKI_multiply2";
  Parameter_Type => (Hw_Long_Integer, Kernel_Time),
  Mechanism => (Value, Reference),
  Result_Type => Kernel_Time);

```

```

--/ SUBPROGRAM NAME:

```

```

--/
--/ DESCRIPTION:
--/ multiplies an integer value by a Kernel_time value returning a
--/ Kernel_time value

```

```

--/ --REQ: 12.1.23, 12.1.24

```

```

--/ --PRIM: N/A

```

```

--/ --TIME: N/A

```

```

--/ PARAMETERS:

```

```

--/ left
--/ left operand of the multiplication operator

```

```

--/ default value:

```

```

--/ none

```

```

--/ right
--/ right operand of the multiplication operator

```

```

--/ default value:

```

```

--/ none

```

```

--/ function return value

```

```

--/ result of the multiplication operator

```

```

--/ PRECONDITIONS:

```

```

--/ NONE

```

```

--/ ACTIONS:

```

```

--/ takes the mathematical product of the integer and the Kernel_time
--/ value, returning a properly formatted Kernel_time result

```

```

--/ POSTCONDITIONS:

```

```

function "/" (
  Left : Kernel_Time;
  Right : Hw_Long_Integer)
return Kernel_Time;

pragma Interface(Assembly, "/");
pragma Import_Function (Internal => "/",
  External => "GKT_divide",
  Parameter_Types => (Kernel_Time, Hw_Long_Integer),
  Mechanism => (Reference_Value),
  Result_Type => Kernel_Time);

```

```

--/-----
--/ SUBPROGRAM NAME:
--/ "/"

```

```

--/ DESCRIPTION:
--/ divides a Kernel_time value by an Integer value returning a
--/ Kernel_time value
--/ --REQ: 12.1.23, 12.1.24
--/ --PRIM: N/A
--/ --TIME: N/A
--/ PARAMETERS:
--/ left left operand of the division operator (dividend)
--/ right right operand of the division operator (divisor)
--/ default value:
--/ none
--/
--/ function return value
--/ result of the division operator
--/ PRECONDITIONS:
--/ NONE
--/ ACTIONS
--/ takes the mathematical quotient of the Kernel_time value and the
--/ Integer, truncating the result toward zero if necessary, returning
--/ a properly formatted Kernel_time result
--/ POSTCONDITIONS:
--/ the quotient of left divided by right is computed and returned
--/ ERROR PROCESSING:
--/ NONE

```

```

--/-----
--/ pragma (page);

```



```

function "+" (
  Left : Kernel_Time)
  return Kernel_Time;

pragma Interface(Assembly, "+");
pragma Import Function (Internal
  External => "GKT add",
  Parameter_Types => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Kernel_Time);

pragma Import Function (Internal
  External => "GKT unary plus",
  Parameter_Types => (Kernel_Time),
  Mechanism => (Reference),
  Result_Type => Kernel_Time);

--/ =====
--/ SUBPROGRAM NAME:
--/ "+"
--/
--/ DESCRIPTION:
--/ unary plus of a Kernel_time value returning a Kernel_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the unary plus operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the unary plus operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical unary plus of the Kernel_time value, returning
--/ a properly formatted Kernel_time result
--/
--/ POSTCONDITIONS:
--/ the unary plus of left is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/ =====
pragma (page);

```

```

function "-" (
  Left : Kernel_Time)
  return Kernel_Time;

pragma Interface(Assembly, "-");
pragma Import Function (Internal
  External => "GKT subtract",
  Parameter_Types => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Kernel_Time);

pragma Import Function (Internal
  External => "GKT unary minus",
  Parameter_Types => (Kernel_Time),
  Mechanism => (Reference),
  Result_Type => Kernel_Time);

--/ =====
--/ SUBPROGRAM NAME:
--/ "-"
--/
--/ DESCRIPTION:
--/ unary negation of a Kernel_time value returning a Kernel_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the unary negation operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the negation operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical unary negation of the Kernel_time value,
--/ returning a properly formatted Kernel_time result
--/
--/ POSTCONDITIONS:
--/ the unary negation of left is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/ =====
pragma (page);

```

```

function "<" (
    Left : Kernel_Time;
    Right : Kernel_Time)
return Boolean;

pragma Interface(Assembly, "<");
pragma Import Function (Internal => "<",
    External => "GKT_Is",
    Parameter_Type => (Kernel_Time, Kernel_Time),
    Mechanism => (Reference, Reference),
    Result_Type => Boolean);

--/ SUBPROGRAM NAME:
--/ "<"
--/
--/ DESCRIPTION:
--/ compares two Kernel time values returning a Boolean true if left is
--/ less than right, false otherwise
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the less-than operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the less-than operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the less-than operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel time values, returning Boolean true if the
--/ left operand is less than the right, returning false otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```

--/ pragma (page);

```

function "<=" (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Boolean;
--/
--/
pragma (page);

```

```

pragma Interface(Assembly, "<=");
pragma Import Function (Internal
  External => "GKI_le",
  Parameter_Type => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Boolean);

```

```

--/ SUBPROGRAM NAME:
--/ "<="
--/

```

```

--/ DESCRIPTION:
--/ compares two Kernel time values returning a Boolean true if left is
--/ less than or equal to right, false otherwise
--/

```

```

--/ --REQ: 12.1.23, 12.1.24
--/

```

```

--/ --PRIM: N/A
--/

```

```

--/ --TIME: N/A
--/

```

```

--/ PARAMETERS:
--/

```

```

--/ left
--/ left operand of the less-than-or-equal-to operator
--/

```

```

--/ default value:
--/ none
--/

```

```

--/ right
--/

```

```

--/ right operand of the less-than-or-equal-to operator
--/

```

```

--/ default value:
--/ none
--/

```

```

--/ function return value
--/

```

```

--/ result of the less-than-or-equal-to operator
--/

```

```

--/ PRECONDITIONS:
--/

```

```

--/ NONE
--/

```

```

--/ ACTIONS:
--/

```

```

--/ compares the two Kernel time values, returning Boolean true if the
--/ left operand is less than or equal to the right, returning false
--/ otherwise
--/

```

```

--/ POSTCONDITIONS:
--/

```

```

--/ the values of left and right are compared
--/

```

```

--/ ERROR PROCESSING:
--/

```

```

--/ NONE

```

```

--/
pragma (page);

```

```

function ">"(
  Left : Kernel Time;
  Right : Kernel Time)
return Boolean;

pragma Interface(Assembly, ">");
pragma Import Function Internal ">"
  External ">" GKT gr;
  Parameter Types => (Kernel Time, Kernel Time),
  Mechanism      => (Reference, Reference),
  Result_Type    => Boolean);

--/
SUBPROGRAM NAME:
--/ ">"
--/
--/ DESCRIPTION:
--/ compares two Kernel Time values returning a Boolean true if left is
--/ greater than right, false otherwise
--/
--/ --REQ: 12.1.23, 2.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the greater-than operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the greater-than operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the greater-than operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel Time values, returning Boolean true if the
--/ left operand is greater than the right; returning false otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```

```

function ">=" (
  Left : Kernel Time;
  Right : Kernel Time)
  return Boolean;

pragma Interface(Assembly, ">=");
pragma Import Function (Internal => ">=",
  External -- "GKT.ge",
  Parameter, y_pos => (Kernel_Time, Kernel_Time),
  Mechanism --> (Reference, Reference),
  Result_Type --> Boolean);
end Generic Kernel Time;
pragma (page);

```

```

--/ =====
--/ SUBPROGRAM NAME:
--/ ">="
--/
--/ DESCRIPTION:
--/ compares two Kernel time values returning a Boolean true if left is
--/ greater than or equal to right, false otherwise
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the greater-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the greater-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the greater-than-or-equal-to operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel time values, returning Boolean true if the
--/ left operand is greater than or equal to the right, returning
--/ false otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE

```

```

--/ .....
--/ MODULE NAME:
--/   Kernel_time
--/
--/ MODULE TYPE:
--/   package specification
--/
--/ MODULE PURPOSE:
--/   provide the abstraction of Kernel time
--/
--/ .....
--/ MODULE DESCRIPTION:
--/   Kernel_time is a hand instantiation of package generic kernel_time
--/
--/ .....
--/ MODULE CONTENTS:
--/   Kernel_time (generic package specification)
--/   Integral_duration (type)
--/   Kernel_time (type)
--/   seconds (overloaded (2) function)
--/   milliseconds (function)
--/   microseconds (function)
--/   ticks_per_second (constant)
--/   ticks_per_second_value (generic formal parameter)
--/   zero_Kernel_time (constant)
--/   "+" (unary: function)
--/   "*" (function)
--/   "-" (unary: function)
--/   "/" (function)
--/   "<=" (function)
--/   ">=" (function)
--/   "<" (function)
--/   ">" (function)
--/
--/ REFERENCES:
--/
--/ DESIGN DOCUMENTS:
--/   Kernel_Facilities_Definition_Document
--/   --REQ: 12.1.1, 12.1.5, 12.1.6, 12.1.25
--/   --PRIM: N/A
--/   --TIME: N/A
--/
--/ USER'S MANUAL:
--/   Kernel_User's_Manual
--/
--/ TESTING AND VALIDATION:
--/   Unit_tests_and_Integration_testing_with_Time_keeper
--/
--/ NOTES:
--/   this package must be instantiated and used in place of the
--/   compiler-provided Package_Calendar
--/
--/   this package provides an abstract data type representing Time
--/   for use by the kernel. It is at the lowest level of a set of
--/   abstractions that jointly respond to Requirements Section 12

```

```

--/ .....
--/ MODIFICATION HISTORY:
--/   18Aug88 bamberg created (compiles clean)
--/   18Oct88 firth revised to interface to assembler
--/   18Oct88 firth changed type of ticks_per... to long int
--/   24Oct88 firth added 'milliseconds' and 'microseconds'
--/   09Jan89 firth improved documentation
--/   11Jan89 firth final doc. update for beta release
--/   16Feb89 firth corrected error in documentation
--/   20Jun89 dvk shifted to VMS definitions
--/   18Jul89 rlv HAND INSTANTIATION due to DEC Ada limitation!!!
--/   generics MAY NOT have MACHINE CODE BODIES!
--/   removed dependency on package system
--/   fixed interface spec for GKT_seconds!
--/
--/ .....
--/ VERSION: 3.0
--/
--/ DISCLAIMER:
--/
--/   The following copyright must be included in this software and
--/   all software utilizing this software.
--/
--/ DISTRIBUTION:
--/
--/   Approved for public release; distribution is unlimited.
--/
--/ Copyright (c) 1989 by Carnegie Mellon University, Pittsburgh,
--/   PA. The Software Engineering Institute (SEI) is a federally
--/   funded research and development center established and operated
--/   by Carnegie Mellon University (CMU). Sponsored by the U.S.
--/   Department of Defense under contract F19628-85-C-0003, the SEI
--/   is supported by the services and defense agencies, with the U.S.
--/   Air Force as the executive contracting agent.
--/
--/ Permission to use, copy, modify, or distribute this software and
--/   its documentation for any purpose and without fee is hereby
--/   granted, provided that the above copyright notice appear in all
--/   copies and that both that copyright notice and this permission
--/   notice appear in supporting documentation. Further, the names
--/   Software Engineering Institute or Carnegie Mellon University may
--/   not be used in advertising or publicity pertaining to
--/   distribution of the software without specific, written prior
--/   permission. CMU makes no claims or representations about the
--/   suitability of this software for any purpose. This software is
--/   provided "as is" and no warranty, express or implied, is made
--/   by the SEI or CMU, as to the accuracy and functioning of the
--/   program and related program material, nor shall the fact of
--/   distribution constitute any such warranty. No responsibility is
--/   assumed by the SEI or CMU in connection herewith.
--/ .....
--/
--/ with Hardware Interface; use Hardware Interface;
--/   to get: hw_duration

```

```

..      hw_long_integer
..      hw_positive
..
pragma (page);

```

```

..      .....
..      package short_name: KT
..      .....
package Kernel_Time
is
..
..      *** BEGIN PSEUDO-GENERIC PARAMETERS ***
..
..      .. the number of Ticks in a second (this value is never used within the
..      .. Kernel except for the initialization of the constant, nor should it be
..      .. used without the Kernel)
..
Ticks_Per_Second_Value : constant Hw_Long_Integer := 500_000;
..
..      *** END PSEUDO-GENERIC PARAMETERS ***
..
..      .. the constant number of Ticks per second (thus providing a mechanism of
..      .. converting from Kernel time to application time)
..
Ticks_Per_Second : Hw_Long_Integer := Ticks_Per_Second_Value;
..
..      type Kernel_time; the time on which elapsed time and epoch time
..      .. abstractions are built
..
..      this time representation allows applications beginning at time zero to
..      .. execute for some 15_000 years, so no overflow is monitored
..
..      .....
..      .. the user should ensure that adjusting any times does not approach the
..      .. limit of this time representation; proper Kernel functioning is NOT
..      .. guaranteed if time overflows
..      .....
..      Kernel_Time is represented as a signed 64-bit binary integer,
..      .. representing a count of hundred-nanoseconds. Hence, a kernel time
..      .. of 10_000_000 corresponds to one second.
..
..      For the purposes of Ada definition, kernel time is a record of
..      .. two components, being respectively the low-order and high-order
..      .. 32 bits. The high-order component can be correctly declared to
..      .. be a signed 32-bit integer, of type hw_long_integer. The low-order
..      .. component, however, is properly an UNSIGNED 32-bit integer, which
..      .. we have defined in HW_INTERFACE as hw_long_unsigned for VAX Ada

```

```

-- Applications should use ONLY the exported constructor functions to
-- create values of type kernel_time or of types derived from it.

```

```

type Kernel_Time is record
  Low : hw_Long_Unsigned := 0;
  High: hw_Long_Integer := 0;
end record;

```

```

for Kernel_Time use record
  Low at 0 range 0..31;
  High at 4 range 0..31;
end record;

```

```

-- the value of zero for the Kernel_time abstraction
Zero_Kernel_Time : constant Kernel_Time := (0,0);

```

```

-- the range of durations represented as integral values

```

```

type Integral_Duration is new hw_Long_Integer;
pragma (page);

```

```

function Seconds (
  Ada_Duration : hw_Duration)
return Kernel_Time;

-- The pragma INTERFACE and IMPORT FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

-- SUBPROGRAM NAME:
-- seconds
-- DESCRIPTION:
-- conversion routine from Ada duration type to Kernel_time
-- REQ: 12.1.23, 12.1.24
-- PRIM: N/A
-- TIME: N/A
-- PARAMETERS:
-- Ada_duration
-- the time to convert in Ada duration format (note that this limits
-- legal values to 0 to 86400 seconds, 1 day)
-- default value:
-- none
-- function return value
-- the input duration in Kernel_time format (note that this contains
-- time up to 1 day)
-- PRECONDITIONS:
-- NONE
-- ACTIONS:
-- returns the Ada duration time in Kernel_time format
-- POSTCONDITIONS:
-- NONE
-- ERROR PROCESSING:
-- NONE
--

pragma (page);

```



```

function Seconds (
  An_Integral_Duration : Integral_Duration
) return Kernel_Time;

pragma Interface(Assembly, Seconds);
pragma Import Function (Internal => Seconds,
  External => "GKT_seconds1",
  Parameter_Types => (hw, Duration),
  Mechanism => (Reference),
  Result_Type => Kernel_Time);

pragma Import Function (Internal => Seconds,
  External => "GKT_seconds2",
  Parameter_Types => (Integral_Duration),
  Mechanism => (Value),
  Result_Type => Kernel_Time);

```

```

--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ an integral duration
--/ the time to convert in Integer format
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the integral duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/

```

pragma (page);

```

function Milliseconds (
  Integral_Milliseconds : Integral_Duration
) return Kernel_Time;

pragma Interface(Assembly, Milliseconds);
pragma Import Function (Internal => Milliseconds,
  External => "GKT_milliseconds",
  Parameter_Types => (Integral_Duration),
  Mechanism => (Value),
  Result_Type => Kernel_Time);

--/ SUBPROGRAM NAME:
--/ milliseconds
--/
--/ DESCRIPTION:
--/ conversion routine from integral elapsed times to Kernel_time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ integral milliseconds
--/ the duration in milliseconds to convert to kernel_time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input integral duration in Kernel_time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ returns the duration time in Kernel_time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--/

```

pragma (page);

```

function Microseconds (
  Integral_Microseconds : Integral_Duration)
  return Kernel_Time;

pragma Interface(Assembly, Microseconds);
pragma Import_Function (Internal => Microseconds,
  External => "GKT_microseconds",
  Parameter_Types => (Integral_Duration),
  Mechanism => (Value),
  Result_Type => Kernel_Time);

--=====
--/ SUBPROGRAM NAME:
--/ seconds
--/
--/ DESCRIPTION:
--/ conversion routine from Integral elapsed times to Kernel time
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ Integral_microseconds
--/ the duration in microseconds to convert to kernel time
--/
--/ default value:
--/ none
--/
--/ function return value
--/ the input Integral duration in Kernel time format
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS
--/ returns the Integral duration time in Kernel time format
--/
--/ POSTCONDITIONS:
--/ NONE
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--=====
pragma (page);

```

```

function "+" (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Kernel_Time;

-- The pragma INTERFACE and IMPORT_FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(notes))

--=====
--/ SUBPROGRAM NAME:
--/ "+"
--/
--/ DESCRIPTION:
--/ adds two Kernel_time values returning a Kernel_time value
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the addition operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the addition operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the addition operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ takes the mathematical sum of the two Kernel_time values, returning a
--/ properly formatted Kernel_time result
--/
--/ POSTCONDITIONS:
--/ the sum of left and right is computed and returned
--/
--/ ERROR PROCESSING:
--/ NONE
--/
--=====
pragma (page);

```

```
function "" {
  Left : Kernel Time;
  Right : Kernel Time;
  return Kernel_Time;
}
```

```
-- The pragma INTERFACE and IMPORT FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(nodes))
```

```
-- SUBPROGRAM NAME:
```

```
-- DESCRIPTION:
-- subtracts the right Kernel time value from the left Kernel time
-- value returning a Kernel_time value
```

```
-- REQ: 12.1.23, 12.1.24
```

```
-- PRIM: N/A
```

```
-- TIME: N/A
```

```
-- PARAMETERS:
```

```
-- left
-- left operand of the subtraction operator (subtrahend)
```

```
-- default value:
-- none
```

```
-- right
```

```
-- right operand of the subtraction operator (minuend)
```

```
-- default value:
```

```
-- none
```

```
-- function return value
```

```
-- result of the subtraction operator
```

```
-- PRECONDITIONS:
```

```
-- NONE
```

```
-- ACTIONS
```

```
-- takes the mathematical difference of the two Kernel time values by
-- subtracting the right operand from the left operand, returning a
-- properly formatted Kernel time result
```

```
-- POSTCONDITIONS:
```

```
-- the difference of left and right is computed and returned
```

```
-- ERROR PROCESSING:
```

```
-- NONE
```

```
pragma (page);
```

```
function "" {
  Left : Kernel Time;
  Right : Hw Long Integer;
  return Kernel_Time;
}
```

```
-- The pragma INTERFACE and IMPORT FUNCTION can be found following
-- the second declaration of the function (as per VAX RTL 13.9(nodes))
```

```
-- SUBPROGRAM NAME:
```

```
-- DESCRIPTION:
-- multiplies a Kernel time value by an integer value returning a
-- Kernel_time value
```

```
-- REQ: 12.1.23, 12.1.24
```

```
-- PRIM: N/A
```

```
-- TIME: N/A
```

```
-- PARAMETERS:
```

```
-- left
-- left operand of the multiplication operator
```

```
-- default value:
-- none
```

```
-- right
```

```
-- right operand of the multiplication operator
```

```
-- default value:
```

```
-- none
```

```
-- function return value
```

```
-- result of the multiplication operator
```

```
-- PRECONDITIONS:
```

```
-- NONE
```

```
-- ACTIONS:
```

```
-- takes the mathematical product of the Kernel time value and the
-- integer, returning a properly formatted Kernel_time result
```

```
-- POSTCONDITIONS:
```

```
-- the product of left and right is computed and returned
```

```
-- ERROR PROCESSING:
```

```
-- NONE
```

```
pragma (page);
```

```

function *** (
  Left : Hw Long Integer;
  Right : Kernel Time)
return Kernel Time;

pragma Interface(Assembly, ***);
pragma Import Function (Internal => ***
  External => "GKT multiply1";
  Parameter_Types => (Kernel Time, Hw Long Integer),
  Mechanism => (Reference, Value),
  Result_Type => Kernel Time);

pragma Import Function (Internal => ***
  External => "GKT multiply2";
  Parameter_Types => (Hw Long Integer, Kernel Time),
  Mechanism => (Value, Reference),
  Result_Type => Kernel Time);

```

```

-- the product of left and right is computed and returned
--
-- ERROR PROCESSING:
-- NONE
--
-- =====
pragma (page);

```

```

-- =====
-- SUBPROGRAM NAME:
-- ***
--
-- DESCRIPTION:
-- multiplies an integer value by a Kernel_time value returning a
-- Kernel_time value
--
-- REQ: 12.1.23, 12.1.24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the multiplication operator
--
-- default value:
-- none
--
-- right
-- right operand of the multiplication operator
--
-- default value:
-- none
--
-- function return value
-- result of the multiplication operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical product of the integer and the Kernel_time
-- value, returning a properly formatted Kernel_time result
--
-- POSTCONDITIONS:

```

```

function "r" (
  Left: Kernel Time;
  Right: Hw Long Integer)
return Kernel Time;

pragma Interface(Assembly, "r");
pragma Import Function (Internal => "r";
  External => "GKT_divide";
  Parameter Types => (Kernel Time, Hw Long Integer);
  Mechanism => (Reference, Value);
  Result_Type => Kernel Time);

```

pragma (page);

```

-- SUBPROGRAM NAME:
-- "r"
--
-- DESCRIPTION
-- divides a Kernel time value by an Integer value returning a
-- Kernel time value
-- REQ: 12.1.23, 12.1.24
-- --PRIM: N/A
-- --TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the division operator (dividend)
-- default value:
-- none
-- right
-- right operand of the division operator (divisor)
-- default value:
-- none
-- function return value
-- result of the division operator
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical quotient of the Kernel time value and the
-- integer, truncating the result toward zero if necessary, returning
-- a properly formatted Kernel time result
--
-- POSTCONDITIONS:
-- the quotient of left divided by right is computed and returned
--
-- ERROR PROCESSING:
-- NONE

```

```

function "+"(
Left : Kernel Time)
return Kernel Time;

pragma Interface(Assembly, "+");
pragma Import Function (Internal
External => "GKT add",
Parameter Types => (Kernel Time, Kernel Time),
Mechanism => (Reference, Reference),
Result Type => Kernel Time);

pragma Import Function (Internal
External => "GKT unary plus",
Parameter Types => (Kernel Time),
Mechanism => (Reference),
Result Type => Kernel Time);

-- SUBPROGRAM NAME:
-- "+"
--
-- DESCRIPTION:
-- unary plus of a Kernel time value returning a Kernel time value
--
-- REQ: 12.1.23, 12.1.24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the unary plus operator
--
-- default value:
-- none
--
-- function return value
-- result of the unary plus operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical unary plus of the Kernel time value, returning
-- a properly formatted Kernel time result
--
-- POSTCONDITIONS:
-- the unary plus of left is computed and returned
--
-- ERROR PROCESSING:
-- NONE
--
pragma (page);

```

```

function "-"(
Left : Kernel Time)
return Kernel Time;

pragma Interface(Assembly, "-");
pragma Import Function (Internal
External => "GKT subtract",
Parameter Types => (Kernel Time, Kernel Time),
Mechanism => (Reference, Reference),
Result Type => Kernel Time);

pragma Import Function (Internal
External => "GKT unary minus",
Parameter Types => (Kernel Time),
Mechanism => (Reference),
Result Type => Kernel Time);

-- SUBPROGRAM NAME:
-- "-"
--
-- DESCRIPTION:
-- unary negation of a Kernel time value returning a Kernel time value
--
-- REQ: 12.1.23, 12.1.24
--
-- PRIM: N/A
--
-- TIME: N/A
--
-- PARAMETERS:
-- left
-- left operand of the unary negation operator
--
-- default value:
-- none
--
-- function return value
-- result of the negation operator
--
-- PRECONDITIONS:
-- NONE
--
-- ACTIONS:
-- takes the mathematical unary negation of the Kernel time value,
-- returning a properly formatted Kernel time result
--
-- POSTCONDITIONS:
-- the unary negation of left is computed and returned
--
-- ERROR PROCESSING:
-- NONE
--
pragma (page);

```

```
function "<" (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Boolean;
```

```
pragma Interface(Assembly, "<");
pragma Import Function (Internal
  External => "GKT_Is",
  Parameter_Types => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Boolean);
```

```
--/ SUBPROGRAM NAME:
```

```
--/ "<"
```

```
--/ DESCRIPTION:
```

```
--/ compares two Kernel_time values returning a Boolean true if left is
--/ less than right, false otherwise
```

```
--/ PARAMETERS:
```

```
--/ left
--/ left operand of the less-than operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the less-than operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the less-than operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is less than the right; returning false otherwise
```

```
--/ POSTCONDITIONS:
```

```
--/ the values of left and right are compared
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

```
pragma (page);
```

```
function "<=" (
  Left : Kernel_Time;
  Right : Kernel_Time)
return Boolean;

pragma Interface(Assembly, "<=");
pragma Import Function (Internal
  External => "GKT_Le",
  Parameter_Types => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Boolean);
```

```
--/ SUBPROGRAM NAME:
```

```
--/ "<="
```

```
--/ DESCRIPTION:
```

```
--/ compares two Kernel_time values returning a Boolean true if left is
--/ less than or equal to right, false otherwise
```

```
--/ REQ: 12.1.23, 12.1.24
```

```
--/ PRIM: N/A
```

```
--/ TIME: N/A
```

```
--/ PARAMETERS:
```

```
--/ left
```

```
--/ left operand of the less-than-or-equal-to operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ right
```

```
--/ right operand of the less-than-or-equal-to operator
```

```
--/ default value:
```

```
--/ none
```

```
--/ function return value
```

```
--/ result of the less-than-or-equal-to operator
```

```
--/ PRECONDITIONS:
```

```
--/ NONE
```

```
--/ ACTIONS:
```

```
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is less than or equal to the right, returning false
--/ otherwise
```

```
--/ POSTCONDITIONS:
```

```
--/ the values of left and right are compared
```

```
--/ ERROR PROCESSING:
```

```
--/ NONE
```

```

--/
--/
pragma (page);

```

```

function ">" (
  Left : Kernel_Time;
  Right : Kernel_Time)
  return Boolean;

pragma Interface(Assembly, ">");
pragma Import Function (Internal
  External => "GKI_gt",
  Parameter_Type => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Boolean);

--/
--/
--/ SUBPROGRAM NAME:
--/ ">"
--/
--/ DESCRIPTION:
--/ compares two Kernel_time values returning a Boolean true if left is
--/ greater than right, false otherwise
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the greater-than operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the greater-than operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the greater-than operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is greater than the right; returning false otherwise
--/
--/ POSTCONDITION'S:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE
--/

```


--/

pragma (page);

```
function ">=" (
  Left: Kernel_Time;
  Right: Kernel_Time)
return Boolean;

pragma Interface(Assembly, ">=");
pragma Import Function (Internal => ">=";
  External => "GKT_ge";
  Parameter_Types => (Kernel_Time, Kernel_Time),
  Mechanism => (Reference, Reference),
  Result_Type => Boolean);

--/ =====
--/ SUBPROGRAM NAME:
--/ ">="
--/
--/ DESCRIPTION:
--/ compares two Kernel_time values returning a Boolean true if left is
--/ greater than or equal to right, false otherwise
--/
--/ --REQ: 12.1.23, 12.1.24
--/
--/ --PRIM: N/A
--/
--/ --TIME: N/A
--/
--/ PARAMETERS:
--/ left
--/ left operand of the greater-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ right
--/ right operand of the greater-than-or-equal-to operator
--/
--/ default value:
--/ none
--/
--/ function return value
--/ result of the greater-than-or-equal-to operator
--/
--/ PRECONDITIONS:
--/ NONE
--/
--/ ACTIONS:
--/ compares the two Kernel_time values, returning Boolean true if the
--/ left operand is greater than or equal to the right, returning
--/ false otherwise
--/
--/ POSTCONDITIONS:
--/ the values of left and right are compared
--/
--/ ERROR PROCESSING:
--/ NONE
```

and Kernel Time;

424
 Local Process, function 337
 Ls, function 36

Message Attributes Entry, type 347
 Message Attributes Entry Pointer, type 347
 Message Contents Type, type 348
 Message Length Type, type 155
 Message Tag Type, type 155
 Microseconds, function 23, 56, 383, 408
 Milliseconds, function 22, 55, 382, 408
 Monitoring Message Tag Type, type 329
 Monitoring Process Type, type 329
 Multiply1, function 28
 Multiply2, function 30

Name Of, function 212
 Nct Entry, type 104, 112
 Network Configuration, package 110
 Network Globals, package 96
 Network Globals, with 102, 110, 161, 303, 344
 Ng, package 162, 345
 Null Handler Body, procedure 221

Package
 Alarm Management 266
 Cg 162, 345
 Communication Globals 157
 Communication Management 191
 Context Save Area 283
 Hardware Interface 6, 364
 Ig 230, 345
 Interrupt Globals 227
 Interrupt Management 241
 Ke 118, 162, 195, 210, 258, 284, 345
 Kernel Time 45, 404
 Network Configuration 110
 Network Globals 99
 Ng 162, 345
 Process Attribute Modifiers 206
 Process Attribute Readers 217
 Process Managers 151
 Process Managers Globals 134
 Process Table 332, 341
 Processor Management 128
 Ptb 162, 195, 210, 345
 Schedule Types 90
 Semaphore Management 255
 St 162, 195, 210, 258, 345
 Tg 118, 162, 195, 258, 284, 345
 Tg 345
 Time Event Queue 308
 Time Globals 82
 Time Management 281
 Timeslice Management 290
 Tool Interface 357
 Tool Interface Output 359
 Pending Activity Attributes Information, type 323
 Pending Activity Type, type 321
 Preemption, type 87

Priority, type 87
 Procedure
 Adjust Elapsed Time 271
 Allocate Device Receiver 188
 Begin Collection 350
 Bind Interrupt Handler 238
 Cancel Alarm 264
 Cease Collection 352
 Claim 246, 248, 250
 Create Process 147
 Destroy Process Information Record 335
 Die 197
 Disable 233
 Disable Time Slicing 288
 Enable 232
 Enable Time Slicing 287
 Initialization Complete 125
 Initialize Master Processor 121
 Initialize Process Table 336
 Initialize Subordinate Processor 123
 Kill 199
 Null Handler Body 221
 Read Interrupt Table 355
 Read Process Table 354
 Receive Message 177, 180, 184
 Release 252
 Send Message 166
 Send Message And Wait 168, 171, 174
 Set Alarm 260, 262
 Set Process Priority 200
 Set Process Preemption 199
 Set Timeslice 286
 Simulate Interrupt 236
 Synchronize 274, 276, 278
 Wait 201, 203
 Process Attribute Modifiers, package 206
 Process Attribute Readers, package 217
 Process Attributes Entry, type 346
 Process Attributes Entry Pointer, type 346
 Process Attributes Information, type 316
 Process Identifier, type 304
 Process Index Type, type 96
 Process Information Record, type 304, 330
 Process Initialization Status Type, type 312
 Process Managers, package 151
 Process Managers Globals, package 134
 Process Managers Globals, with 139, 303
 Process Name Type, type 132
 Process State, type 88
 Process Table, package 332, 341
 Process Table type 349
 Process Table, with 102, 110, 139, 161, 195, 210, 220, 244, 344
 Process Type, type 331
 Process type, type 313
 Processor Identifier, type 96
 Processor Management, package 128
 Ptb, package 162, 195, 210, 345

Read Clock, function 273
 Read Interrupt Table, procedure 355
 Read Process Table, procedure 354
 Release, procedure 252

Schedule Attributes Information, type 318
 Schedule Types, package 90
 Schedule Types, with 139, 161, 195, 210, 244, 258, 269, 303, 344
 Seconds, function 20, 21, 53, 54, 380, 381, 406, 407
 Seconds1, function 20
 Seconds2, function 21
 Semaphore, type 310
 Semaphore Attributes Information, type 327
 Semaphore Head, type 310
 Semaphore Head Ptr, type 310
 Semaphore Management, package 255
 Send Message, procedure 166
 Send Message And Wait, procedure 168, 171, 174
 Set Alarm, procedure 260, 262
 Set Process Preemption, procedure 199
 Set Process Priority, procedure 200
 Set Timeslice, procedure 286
 Simulate Interrupt, procedure 236
 Size Of Process Table, function 338, 353
 St, package 162, 195, 210, 258, 345
 Subtract, function 26
 Subtype
 Event Identifier 308
 Synchronize, procedure 274, 276, 278
 System, with 6, 364

Tg, package 118, 162, 195, 258, 284, 345
 Tg, package 345
 Time Event Queue, package 308
 Time Globals, package 82
 Time Globals, with 118, 161, 195, 244, 258, 269, 284, 290, 344
 Time Keeper Globals, with 303
 Time Management, package 281
 Timeslice Management, package 290
 To Ada Duration, function 75
 To Elapsed Time, function 74, 78
 To Epoch Time, function 79
 To Hw Bist8, function 15, 374
 To Hw Bist8 Ptr, function 10, 11, 368, 369
 To Hw Bist8 Ptr, function 9, 367
 To Kernel Time, function 76, 77
 To Message Attributes, function 347
 To Process Attributes, function 346
 Tool Interface, package 357
 Tool Interface Attributes Information, type 329
 Tool Interface Globals, with 303, 344
 Tool Interface Output, package 359
 Type
 Acknowledged Message Information 325

Address Registers Area 293
 Adjacent Process Table Entry 331
 Bus Address 95
 Communication Attributes Information 320
 Configuration Table 105, 113
 Context Save Area Contents 287
 Data Registers Area 295
 Device Name Type 132
 Elapsed Time 49, 80
 Epoch Time 49, 80
 Event Information 306
 Floating Point Coprocessor Type 286
 Full Process Id 347
 How To Handle Message Queue Overflow 131
 Hw Address 14, 373
 Hw Bist8 7, 365
 Hw Bist8 Ptr 9, 367
 Hw Bist8 7, 365
 Hw Bist8 Ptr 9, 367
 Hw Duration 6, 364
 Hw Integer 12, 370
 Hw Long Integer 12, 370
 Hw Long Natural 13, 371
 Hw Long Positive 13, 371
 Hw Long Unsigned 371
 Hw Natural 13, 371
 Hw Positive 13, 372
 Hw Short Integer 370
 Hw String 15, 374
 Integral Duration 19, 49, 379, 405
 Interrupt Condition 220
 Interrupt Name 220
 Interrupt Owner 220
 Interrupt Source 221
 Interrupt State 221
 Interrupt Table 349
 Interrupt Table Entry 222
 Interrupt Table Type 224
 Kernel Primitive Name Type 345
 Kernel Time 19, 378, 405
 Message Attributes Entry 347
 Message Attributes Entry Pointer 347
 Message Contents Type 348
 Message Length Type 155
 Message Tag Type 155
 Monitoring Message Tag Type 329
 Monitoring Process Type 329
 Nct Entry 104, 112
 Pending Activity Attributes Information 323
 Pending Activity Type 321
 Preemption 87
 Priority 87
 Process Attributes Entry 346
 Process Attributes Entry Pointer 346
 Process Attributes Information 316
 Process Identifier 304
 Process Index Type 96
 Process Information Record 304, 330
 Process Initialization Status Type 312

Process Name Type 132
Process State 88
Process Table 349
Process Table Entry 331
Process Type 313
Processor Identifier 96
Schedule Attributes Information 318
Semaphore 310
Semaphore Attributes Information 327
Semaphore Head 310
Semaphore Head Ptr 310
Tool Interface Attributes Information 329

Time Keeper Globals 303
Tool Interface Globals 303, 344
Unchecked_Conversion 6, 344, 364

Unary Minus function 35
Unary Plus function 34
Unchecked_Conversion with 6, 344, 364

Well procedure 201, 203
Who Am I function 213
With

Communication Globals 161, 303, 344
Context Save Area 303
Context Switcher Globals 303
Datagram Globals 303
Generic Alarm Management 266
Generic Communication Globals 157
Generic Communication Management 191
Generic Interrupt Globals 227
Generic Interrupt Management 241
Generic Kernel Time 45
Generic Network Globals 90
Generic Process Attribute Modifiers 208
Generic Process Attribute Readers 217
Generic Process Managers 151
Generic Process Managers Globals 134
Generic Process Table 340
Generic Processor Management 128
Generic Queue Manager 303
Generic Schedule Types 90
Generic Semaphore Management 255
Generic Time Globals 82
Generic Time Management 281
Generic Timeslice Management 290
Generic Tool Interface 357

Hardware Interface 17, 49, 85, 94, 102, 110,
118, 131, 138, 155, 161, 210, 220, 230, 280,
303, 344, 378, 402
Interrupt Globals 230, 344
Kernel Exceptions 85, 94, 118, 138, 161, 195,
210, 220, 244, 258, 269, 284, 303, 344
Kernel Time 49, 284, 290, 303
Network Globals 102, 110, 161, 303, 344
Process Managers Globals 139, 303
Process Table 102, 110, 139, 161, 195, 210,
220, 244, 344

Schedule Types 139, 161, 195, 210, 244, 258,
269, 303, 344
System 6, 364
Time Globals 118, 161, 195, 244, 258, 269,
284, 290, 344